Actes des 34es journées francophones d'Ingénierie des Connaissances

Cassia Trojahn

AfIA

Association française
pour l'Intelligence Artificielle

IC

Journées francophones d'Ingénierie des Connaissances

PFIA 2023

Table des matières

Éditorial

Journées francophones d'Ingénierie des Connaissances

Les journées francophones d'Ingénierie des Connaissances (IC) sont organisées chaque année depuis 1997, d'abord sous l'égide du Gracq (Groupe de Recherche en Acquisition des Connaissances) puis sous celle du collège SIC (Science de l'Ingénierie des Connaissances) de l'AFIA. Cette année encore, IC est hébergée par la plateforme PFIA, conjointement avec d'autres conférences francophones dans le domaine de l'intelligence artificielle (IA).

L'ingénierie des connaissances peut être vue comme la thématique de l'Intelligence Artificielle accompagnant l'évolution des sciences et technologies de l'information et de la communication qui engendrent des mutations dans les pratiques individuelles et collectives. Elle ambitionne de contribuer à son essor en développant les modèles, les méthodes et les outils pour l'acquisition, la représentation et l'intégration de connaissances afin de rendre possible leur exploitation dans des environnements informatiques aux caractéritiques variées. La représentation formelle de ces connaissances permet des raisonnements automatiques sur ces connaissances et sur les données qui leur sont associées, pouvant être complexes, hétérogènes et évolutives. Sa finalité est la production de méthodes et outils « intelligents », capables d'aider l'humain dans ses activités et ses prises de décisions.

La conférence IC est un lieu d'échanges et de réflexions, de présentation et de confrontation des théories, pratiques, méthodes et outils autour de l'ingénierie des connaissances. Cette communauté prend désormais en compte l'essor des algorithmes d'apprentissage automatique et leurs retombées sur les pratiques individuelles et collectives, tout en conservant l'humain au centre des systèmes de décision exploitant les données et les connaissances. Cette année, les propositions portant sur le thème « apports de graphes de connaissances pour les approches neuro-symboliques d'apprentissage dans l'ingénierie des connaissances » ont été particulièrement bienvenues.

Cette année, la conférence IC a reçu 32 soumissions d'articles : 13 articles longs, 11 articles courts, 2 posters, 1 article de positionnmenet et 5 articles déjà publiés dans une conférence ou revue internationale de renom. Grâce au travail conséquent des membres du comité de programme, chaque article a reçu entre 3 et 4 relectures comportant des critiques argumentées et constructives pour les auteurs. Sur la base de ces critiques, le comité de programme, qui s'est réuni en distanciel, a sélectionné 7 articles longs et 10 articles courts, 1 article de positionnement et 1 poster. Il a également retenu 3 articles de travaux déjà publiés et résumés en Français. Les auteurs de 5 articles ont été invités à soumettre des démonstrations. Cette année, des démonstrations IC ont été invitées à candidater sur le prix PFIA (ensemble des conférences) du meilleur démonstrateur.

Le programme de la conférence réparti sur 3 jours est organisé en 7 sessions dont le contenu est détaillé dans ces actes. Ces sessions portent sur des thèmes qui sont au coeur de l'ingénierie des connaissances tels que « extraction d'informations et graphes de connaissances », « alignement d'ontologies et liage de données », « ontologies et raisonnement pour les systèmes complexes ». D'autres sessions concernent des thèmes émergeants dans la communauté tels que « connaissances, apprentissage, temporalité » et « ingénierie de connaissances, données FAIR ».

Pour cette édition 2023 de la conférence, nous avons l'honneur d'accueillir deux conférences invitées : Pascal Hitzler – Professor Creativity in Engineering Chair and Director of the Center for Artificial Intelligence and Data Science, Department of Computer Science at Kansas State University – dont la conférence invitée est intitulée « Knowledge graphs in neurosymbolic learning approaches » et Heiko Paulheim – Pr. Dr., Data Science, University of Mannheim, Allemagne –, dont la conférence invitée est intitulée « Knowledge graph embedding for data mining with RDF2vec ».

Je voudrais remercier chaleureusement les membres du comité de programme de leur très forte implication, ce qui a oeuvré pour le succès de cette édition 2023 de la conférence IC. Je adresse également de vifs remerciements à l'ensemble des acteurs de la communauté francophone d'Ingénierie des Connaissances qui ont contribué au succès d'IC 2023, ainsi que le comité d'organisation de la plateforme PFIA 2023 qui a été d'une aide précieuse.

Cassia Trojahn

Comité de programme

Présidence

— Cassia Trojahn - IRIT, Université de Toulouse, IRIT.

Membres

— Nathalie Abadie, IGN/COGIT ;
— Marie-Helene Abel, Université de Technologie de Compiègne ;
— Mehwish Alam, Télécom Paris ;
— Xavier Aimé, Cogsonomy ;
— Yamine Ait Ameur, Université de Toulouse, IRIT ;
— Nathalie Aussenac-Gilles, Université de Toulouse, IRIT ;
— Bruno Bachimont, University de technologie de Compiègne ;
— Nacéra Bennacer, Centrale Supélec ;
— Nathalie Bricon-Souf, Université de Toulouse, IRIT ;
— Sandra Bringay, LIRMM - Université Paul Valéry ;
— Patrice Buche, INRAE ;
— Davide Buscaldi, École Polytechnique ;
— Sylvie Calabretto, INSA de Lyon ;
— Pierre-Antoine Champin, ERCIM ;
— Jean Charlet, AP-HP & INSERM UMRS 1142 ;
— Victor Charpenay, Mines Saint-Etienne ;
— Jérôme David, INRIA & Université Grenoble Alpes ;
— Sylvie Despres, LIMICS - Université Sorbonne Paris Nord ;
— Gayo Diallo, Université de Bordeaux ;
— Gilles Falquet, University of Geneva ;
— Catherine Faron, Université Côte d'Azur ;
— Béatrice Fuchs, LIRIS - université de Lyon ;
— Frédéric Fürst, MIS - Université de Picardie ;
— Alban Gaignard, CNRS ;
— Jean-Gabriel Ganascia, Pierre and Marie Curie University - LIP6 ;
— Ollivier Haemmerlé, Université de Toulouse, IRIT ;
— Mounira Harzallah, LS2N - University of Nantes ;
— Nathalie Hernandez, Université de Toulouse, IRIT ;
— Dominique Lenne Heudiasyc, Université de Technologie de Compiègne ;
— Liliana Ibanescu, AgroParisTech ;
— Sébastien Iksal, LIUM - Le Mans Université ;
— Antoine Isaac, Europeana & VU University Amsterdam ;
— Khadija Jradeh, Université de Toulouse, IRIT ;
— Clement Jonquet, MISTEA (INRAE) and LIRMM (U. Montpellier) ;
— Mouna Kamel, Université de Toulouse, IRIT ;
— Gilles Kassel, University of Picardie Jules Verne ;
— Michel Leclère, University of Montpellier (LIRMM/INRIA) ;
— Maxime Lefrançois, MINES Saint-Etienne ;
— Pascal Molli, University of Nantes ;
— Jérôme Nobécourt, LIPN - Université Sorbonne Paris Nord ;
— Nathalie Pernelle, LIPN - Université Sorbonne Paris Nord ;
— Yannick Prié, LINA - University of Nantes ;
— Cédric Pruski, Luxembourg Institute of Science and Technology ;
— Joe Raad, University of Paris-Saclay ;
— Sylvie Ranwez, LGI2P - Ecole des Mines d'Alès ;
— Catherine Roussey, INRAE ;
— Pascal Salembier, UTT ;
— Fatiha Saïs, LISN, CNRS & Université Paris Saclay ;
— Karim Sehaba, LIRIS CNRS ;
— Danai Symeonidou, INRAE ;

— Konstantin Todorov, LIRMM ;
— Rallou Thomopoulos, INRAE ;
— Raphaël Troncy, EURECOM ;
— Haifa Zargayouna, Université Sorbonne Paris Nord.

Session 1 : Extraction d'informations et graphes de connaissances

Création d'un graphe de connaissances géohistorique à partir d'annuaires du commerce parisien du 19ème siècle: application aux métiers de la photographie

S. Tual[1], N. Abadie[1], B. Duménieu[2], J. Chazalon[3], E. Carlinet[3]

[1] LASTIG, Univ. Gustave Eiffel, IGN-ENSG
[2] CRH, EHESS
[3] LRDE, EPITA

solenn.tual@ign.fr; nathalie-f.abadie@ign.fr; bertrand.dumenieu@ehess.fr; joseph.chazalon@epita.fr; edwin.carlinet@epita.fr

Résumé

Les annuaires professionnels anciens, édités à un rythme soutenu dans de nombreuses villes européennes tout au long des XIX^{e} et XX^{e} siècles, forment un corpus de sources unique par son volume et la possibilité qu'ils donnent de suivre les transformations urbaines à travers le prisme des activités professionnelles des habitants, de l'échelle individuelle jusqu'à celle de la ville entière. L'analyse spatio-temporelle d'un type de commerces au travers des entrées d'annuaires demande cependant un travail considérable de recensement, de transcription et de recoupement manuels. Pour pallier cette difficulté, cet article propose une approche automatique pour construire et visualiser un graphe de connaissances géohistorique des commerces figurant dans des annuaires anciens. L'approche est testée sur des annuaires du commerce parisien du XIX^{e} siècle allant de 1799 à 1908, sur le cas des métiers de la photographie.

Mots-clés

Graphe de connaisssances géohistorique, annuaires anciens, reconnaissance et résolution d'entités nommées, bruit OCR, visualisation spatio-temporelle.

Abstract

Business directories have been published at a high frequency in many European cities throughout the 19th and 20th centuries. This corpus of historical sources is unique because of its volume and the opportunity it gives to follow urban transformations through the professional activities of the inhabitants, from the individual scale to that of the entire city. However, the spatio-temporal analysis of businesses of a given type through directory entries requires a considerable amount of manual work. To overcome this difficulty, this article proposes an automatic approach to construct and visualise a geohistorical knowledge graph of businesses listed in old directories. The approach is tested on 19th century Parisian trade directories from 1799 to 1908, on the case of photographers.

Keywords

Geohistorical knowledge graph, old directories, named entity recognition and linking, OCR noise, spatio-temporal visualization.

1 Introduction

A partir de la fin du XVIII^{ème} siècle, les annuaires des habitants et des commerces (voir figure 1), sortes de "pages blanches" et "pages jaunes" avant l'heure, ont connu un succès croissant et ont été édités pour de nombreuses villes européennes et nord-américaines. Ils recensent les habitants, leurs activités professionnelles et leurs localisations, et constituent des sources historiques extrêmement riches pour suivre les évolutions urbaines, de l'échelle individuelle à celle de la ville. Ainsi, [3] évalue le potentiel des annuaires des habitants de Berlin de 1880 pour réaliser des études socio-économiques et démographiques en l'absence de données de recensement. Cette étude, réalisée pour une unique date, démontre l'utilisabilité des entrées extraites automatiquement et suggère, en perspectives, de lier les entrées similaires d'une édition à l'autre et d'étendre les analyses aux annuaires du commerce. [4] utilise des annuaires parus entre 1936 et 1990 pour localiser les anciennes stations services de la ville de Providence (Rhode Island) aux Etats-Unis, afin de détecter des zones potentiellement polluées. L'approche proposée par [4] comprend plusieurs étapes, qu'elle vise à automatiser le plus possible : analyse de la mise en page des annuaires, reconnaissance optique de caractères (OCR), reconnaissance des entités nommées, réalisée ici à l'aide de patrons lexico-syntaxiques, et géocodage à l'aide d'une base d'adresses récente. Le fait de retrouver une même station service dans plusieurs annuaires successifs n'ayant pas d'intérêt pour l'application visée, ce travail n'aborde pas la question du liage des entrées d'annuaires successifs.

Dans cet article, nous proposons d'adapter et d'étendre cette approche pour construire et peupler un graphe de connaissances géohistorique permettant de suivre l'évolu-

tion d'un commerce au cours du temps. L'objectif est de se doter de données structurées, permettant le suivi individuel et collectif des commerces d'un type donné au cours du temps et dans l'espace parisien ancien. Nous la mettons en oeuvre sur les entrées relatives aux métiers de la photographie, mais elle peut être appliquée à n'importe quelle activité professionnelle représentée dans ces annuaires.

L'article est organisé de la façon suivante : la section 2 présente les travaux antérieurs sur la création de graphes géohistoriques ; la section 3 décrit les questions de compétences associées à notre graphe de connaissances ; la section 4 détaille les étapes de création du graphe ; la section 5 propose une application de visualisation spatio-temporelle du graphe et l'évaluation des questions de compétence ; la section 6 discute des perspectives de ce travail.

2 Travaux antérieurs

La création d'un graphe de connaissances à partir d'annuaires anciens nécessite d'en extraire et structurer les informations textuelles. Cette section passe en revue les travaux relatifs à ces différentes étapes : extraction du texte, reconnaissance et liage des entités nommées.

2.1 Détection de mise en page et OCR

Les approches d'analyse de mise en page visent à identifier et étiquetter les régions homogènes de documents. Dans son état de l'art, [7] distingue trois stratégies. Les stratégies descendantes ou *top-down*, comme XY-cut [20] et ses dérivées [11, 26], appliquent des règles pour divisiser progressivement le document en portions de plus en plus petites, jusqu'à atteindre un critère d'arrêt prédéfini ou bien qu'il ne soit plus possible de créer de portion plus petite. Les stratégies ascendantes ou *bottom-up*, comme la méthode docstrum [22] ou les approches par apprentissage automatique, partent des portions élémentaires de documents (des pixels, des mots, etc.) et les regroupent pour créer des régions homogènes, jusqu'à atteindre un critère d'arrêt prédéfini. Enfin, les approches hybrides combinent des techniques ascendantes et descendantes. Plus récemment, l'essor des réseaux de neurones de type transformer a conduit à la proposition de nombreuses approches multimodales comme LayoutLM [29] et ses variantes comme [16]. Elles tirent parti des informations textuelles, visuelles et spatiales des documents pour reconnaître des mises en pages très complexes et variées, mais nécessitent des ressources importantes pour être mises en oeuvre.

Les systèmes d'OCR récents, comme Tesseract [25], OCRopus [8], Kraken [13], Calamari [28] ou Pero OCR [14] s'appuient sur des architectures à base de réseaux de neurones convolutifs (CNN) et de réseaux *Long short-term memory* (LSTM). Ils obtiennent globalement de bons résultats sur des textes récents, mais sur les textes anciens, pour lesquels moins de données d'entraînement sont disponibles, leurs performances baissent. Pour pallier cette difficulté, Pero OCR intègre une couche pour détecter le style de transcription le plus adapté au texte à traiter [14].

2.2 Reconnaissance d'entités nommées

De nombreuses approches ont été proposées pour localiser et classer les portions de texte qui désignent des entités de types prédéfinis comme des personnes, des lieux ou des organisations [19]. Les approches à base de règles utilisent des patrons lexico-syntaxiques combinant catégories grammaticales et entrées de dictionnaires [4, 18]. Sur des corpus spécialisés, lorsque l'on dispose de dictionnaires exhaustifs, elles produisent de bons résultats, mais l'élaboration des patrons constitue un effort important. Les approches supervisées regroupent les techniques d'apprentissage statistique traditionnel et les techniques à base de réseaux de neurones profonds. Comme les approches par patrons, les premières exploitent des descripteurs textuels choisis par un expert. Les secondes, en revanche, définissent leurs propres descripteurs pour classer les tokens selon leur appartenance à un type d'entités nommées. Les modèles de langue récents peuvent être adaptés à des corpus spécialisés avec relativement peu de données d'entraînement et sont très susceptibles de produire les meilleurs résultats [17].

2.3 Construction de graphes géohistoriques et liage de ressources

De nombreux modèles ont été proposés pour représenter des données spatio-temporelles [24]. Les travaux récents sur la représentation des états passés successifs du territoire, reposent majoritairement sur des modèles de graphes. Ainsi [6, 15, 5] s'inspirent du modèle de graphe spatio-temporel de [10] ; dans le premier cas, il s'agit de rues de Paris vectorisées à partir de plans à grande échelle levés à différentes périodes du XIX^{e} siècle, dans le second, des parcelles agricoles issues de plusieurs millésimes du Registre Parcellaire Graphique [1], et dans le troisème, d'unités territoriales statistiques produites par Eurostat et d'unités administratives suisses produites par Swisstopo. Ce dernier travail utilise les standards du Web de données pour représenter et publier les graphes créés. Ces trois approches de construction de graphes géohistoriques utilisent des séries temporelles de données géographiques dont elles extraient les relations spatio-temporelles à l'aide de méthodes de liage entre états sucessifs des entités géographiques considérées. Les approches de liage de données visent à créer des liens de correspondance explicites entre ressources représentant une même entité du monde réel, éventuellement à des temporalités différentes. [23] distingue deux principales catégories de méthodes de liage. Les méthodes fondées sur les données reposent sur l'hypothèse selon laquelle deux ressources présentant des valeurs similaires pour leurs propriétés similaires sont très suceptibles de représenter une même entité du monde réel. C'est le type d'approche mis en oeuvre par des outils comme Silk [2][12] ou LIMES [3][21]. Les méthodes fondées sur les connaissances exploitent les connaissances fournies par l'ontologie qui décrit les données. Les restrictions désignant des ensembles de propriétés

1. Voir : https ://geoservices.ign.fr/rpg
2. http ://silkframework.org/
3. http ://aksw.org/Projects/LIMES.html

comme clés d'identification de ressources sont particulièrement utilisées par ces approches. De nombreux travaux sont ainsi dédiés à l'identification des clés pour le liage, comme [27] ou [2]. Les approches proposées par [6, 15, 5] appartiennent à la première catégorie. Elles reposent essentiellement sur l'évaluation de la similarité de la forme et de la localisation des entités géographiques à lier.

3 Questions de compétence

Ce travail vise à adapter et étendre la chaîne de traitement proposée par [4] pour construire un graphe de connaissances géohistorique à partir d'annuaires anciens. L'objectif de ce modèle de connaissances est d'aider les historiens à suivre et analyser les évolutions des commerces sur le territoire considéré. Ces évolutions peuvent porter sur la nature même des commerces, sur leurs localisations, sur leur pérénité, sur leurs modes d'organisation, etc. Nous avons donc retenu les questions de compétences suivantes, définies avec les historiens du projet. Il s'agit des questions auxquelles on souhaite a minima pouvoir répondre, et que nous supposons suffisament générales pour pouvoir s'appliquer à la plupart des types de commerces figurant dans les annuaires.

CQ1. Quelle est l'adresse du commerce X en 1861 ?
CQ2. Combien y a-t-il de commerces de ce type localisés rue de Rivoli en 1856 ?
CQ3. Quels sont les commerces situés dans une zone définie par un polygone ou un rectangle englobant en 1875 ?
CQ4. Quels commerces ont déménagé au cours de leur existence ?
CQ5. Quels commerces ont été repris par un autre commerçant exerçant la même activité ?

Par ailleurs, les logiques d'organisation spatio-temporelles des commerces peuvent être difficiles à mettre en évidence à l'aide de simples requêtes et nécessitent souvent des analyses spatio-temporelles plus complexes. Par exemple, identifier la multiplication de commerces du même type tenus par les membres d'une même famille dans un même quartier exige d'expliciter à la fois les liens familiaux entre les propriétaires de commerces, la proximité spatiale des commerces sur une période donnée et d'éventuelles logiques de transmissions intra-familiales. Pour faciliter ce type d'analyses complexes, nous proposons donc d'accompagner notre graphe de connaissance géohistoriques d'une application de visualisation spatio-temporelle des données.

4 Construction du graphe de connaissances géohistorique

Les informations contenues dans les annuaires peuvent être vues comme des séries temporelles de données semi-structurées sur les commerces qu'elles décrivent. Nous proposons donc une approche d'extraction d'informations et de construction de graphe de connaisances qui reprend et adapte les étapes de la chaîne de traitement de [4] et les approches à base de liage de [6], [15] et [5].

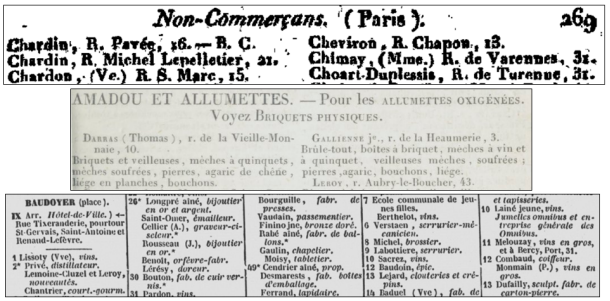


FIGURE 1 – Exemples de mises en pages et d'index différents dans les annuaires *En haut* : Duverneuil et La Tynna 1806 - index par noms ; *Au milieu* : Deflandre 1828 - index par professions ; *En bas* : Bottin 1851 - index par rues.

4.1 Les annuaires du commerce parisien

Le corpus utilisé rassemble des annuaires publiés annuellement entre 1799 et 1908 par différents éditeurs et couvrant 88 années. Leurs contenus varient donc d'une édition à l'autre, en termes d'informations disponibles, d'organisation (index par noms, rues ou professions), de mise en page, de police d'écriture, etc. (voir Figure 1). Ils sont conservés dans différentes bibliothèques parisiennes et ont été scannés indépendamment les uns des autres, avec des niveaux de qualité variables. Les entrées des index par noms comportent généralement le nom du commerce ou de son propriétaire, le type d'activité exercée, d'éventuels titres honorifiques ou médailles professionnelles, le nom de la rue et le numéro et éventuellement une précision sur le type du local, comme "atelier", "entrepôt" ou "boutique", lorsque plusieurs adresses sont fournies. L'entrée de l'annuaire Didot-Bottin de 1860 "Aubert (Mme), couturière, Guénégaud, 10" est un exemple typique d'entrée des index par noms.

4.2 Segmentation de mise en page et OCR

Les annuaires à traiter présentent différentes mises en pages selon les éditions et selon les index. Cependant, celles-ci restent relativement homogènes : les entrées sont toujours organisées en colonnes (de 1 à 5 selon les éditions) et éventuellement séparées par des titres. Le choix a donc été fait de mettre en oeuvre une approche hybride à base de techniques classiques de nettoyage des scans et d'analyse de leurs mises en page pour détecter les entrées : 1) XY-cuts et classification de régions, 2) Détection des lignes (watershed), 3) Regroupement des lignes en entrées. Sur notre corpus, ces techniques s'avèrent extrêmement performantes et peu coûteuses à mettre en oeuvre.

Enfin, pour extraire le texte de chaque entrée, nous avons utilisé la version "sur étagère" de l'outil Pero OCR.

4.3 Reconnaissance des entités nommées

Si les éléments constitutifs des entrées d'annuaires restent globalement les mêmes, leur présentation, en revanche, varie d'une édition à l'autre. A celà, s'ajoutent les erreurs de l'OCR. Les approches de reconnaissance d'entités nommées à base de règles semblent donc inappropriées, car elle

nécessiteraient de définir un nombre de règles trop important pour gérer tous les cas.
Nous avons donc adopté une approche supervisée, et adapté un réseau de neurones profond de reconnaissance d'entités nommées utilisant le modèle de langue CamemBERT pour traiter notre corpus. Nous avons procédé à un pré-entrainement non-supervisé sur plusieurs milliers de pages d'annuaires et à un entraînement supervisé sur un corpus annoté avec les types d'entités que l'on cherche à reconnaître : PER pour les noms de commerces ou de personnes, ACT pour le type d'activité, LOC pour les noms de rues, CARDINAL pour les numéros, TITRE pour les distinctions et FT pour les précisions sur les adresses. Pour limiter les effets négatifs des erreurs d'OCR sur les résultats de reconnaissance des entités nommées, nous avons entraîné le modèle sur du texte bruité. Sur notre corpus de test, de 1669 entrées, également bruitées, le modèle obtient ainsi un score de F-mesure globale de 94.1%. Les étapes d'extraction du texte et de reconnaissance des entités nommées et les résultats obtenus sont décrits en détail dans [1].

4.4 Géocodage historique des entrées

La mise en oeuvre des deux étapes précédente a permis de construire une base de données comportant 9 821 898 entrées. Pour doter les entrées d'annuaires de coordonnées, nous avons procédé au géocodage des adresses. Nous avons utilisé le géocodeur historique développé par le groupe de travail Geohistoricaldata [4]. Sa base de données d'adresses a été saisie à partir de différents plans de Paris du XIX^{e} siècle et l'outil favorise les adresses issues de plans dont la date de production est proche de celle des données à géocoder [5].

4.5 Sélection et représentation des entrées en RDF

Pour faciliter la suite du traitement, nous proposons de filtrer les entrées pour ne conserver que celles concernant le type de commerces à étudier. Dans le cas des métiers de la photographie, nous nous sommes appuyés sur une liste de 252 photographes parisiens extraite de l'ouvrage de [9], qui couvre la période 1820-1910. Nous avons recherché les entrées associées à ces photographes et recencé les activités mentionnées dans ces entrées pour en retenir trois mots-clés que l'on suppose représentatifs des entrées décrivant des photographes : *photo*, *daguer* et *opti*. Puis nous avons converti en RDF et exporté les 34 062 entrées dont l'attribut "activité" comportait ces mots-clés, à l'aide d'un script R2RML. 26 275 d'entre elles ont pu être géocodées à l'étape précédente, soit environ 70% du jeu de données.

4.6 Liage des entrées

Deux méthodes de liage des entrées ont été implémentées. La première méthode génère les liens par inférences en utilisant les clés déclarées dans l'ontologie qui décrit les données. Les propriétés qui composent les clés sont identifiées à l'aide de Sakey [27] : (1) le numéro de l'entrée, (2) le nom et l'adresse, (3) le nom et l'activité et (4) l'activité et l'adresse. Les clés 2 à 4 sont des 1-quasi-clés identifiées sur les données d'un annuaire. On tolère donc une exception afin de gérer l'existence possible de deux index dans le même annuaire. La propriété adresse est créée par concaténation des valeurs des entités de type LOC et CARDINAL, préalablement à l'exécution de Sakey. Tous les caractères ont été passés en minuscules et les éléments de ponctuation situés en début et fin de chaînes ont été supprimés.
La seconde méthode est fondée sur les données. Elle exploite la similarité des valeurs des propriétés des ressources. Elle est mise en oeuvre avec Silk. Après suppression des caractères spéciaux et passage des caractères en minuscule, la distance d'édition Token-Wise est calculée pour les propriétés nom, activité et adresse de chaque paire de ressources. Le score de similarité associé à l'adresse est produit à l'aide d'une moyenne pondérée des résultats obtenus pour les valeurs de LOC et CARDINAL. Enfin, pour les combinaisons de propriétés suivantes - Nom et Activité, Nom et Adresse, Activité et Adresse - le score agrégé retenu correspond à la valeur de la distance Token-Wise la plus faible obtenue par l'une des propriétés de la combinaison. Finalement, seuls les liens dont le score est supérieur à 0.8 sont conservés. 250 622 liens *owl :sameAs* ont été créés avec la méthode fondée sur les connaissances et 357 130 avec la méthode fondée sur les données. Le nombre total des liens calculés et inférés est finalement de 401 852 liens distincts.

5 Visualisation et évaluation

Nous avons évalué le graphe de deux façons. D'une part, nous avons traduit les questions de compétences en requêtes SPARQL, afin de nous assurer que nous obtenions les réponses attendues. Le graphe est accessible ici : `https://dir.geohistoricaldata.org/`. D'autre part, nous avons développé une application de visualisation spatio-temporelle, qui permet d'analyser visuellement les données, sans avoir à écrire de requêtes.

5.1 Evaluation des questions de compétences

Ainsi, notre première question de compétence peut se vérifier avec la requête suivante, qui renvoie "11 rue suger" :

```
PREFIX locn : <http ://www.w3.org/ns/locn#>
PREFIX ont : <http ://rdf.geohistoricaldata.org/def/directory#>
PREFIX rdfs : <http ://www.w3.org/2000/01/rdf-schema#>
PREFIX prov : <http ://www.w3.org/ns/prov#>
PREFIX pav : <http ://purl.org/pav/>
SELECT ?fullAdd
WHERE { ?e a ont :Entry.
?e rdfs :label ?label.
?e prov :wasDerivedFrom ?directory.
?directory pav :createdOn "1861"@fr.
?e locn :address ?add.
?add locn :fullAddress ?fullAdd.
Filter regex( ?label, "gallino").}
```

Les requêtes SPARQL correspondant aux autres questions de compétences listées sont fournies, avec l'en-

4. `https://api.geohistoricaldata.org/docs/#/Geocoding`
5. Ce jeu de données géocodées est publié sur le dépôt suivant : `https://nakala.fr/10.34847/nkl.98eem49t`

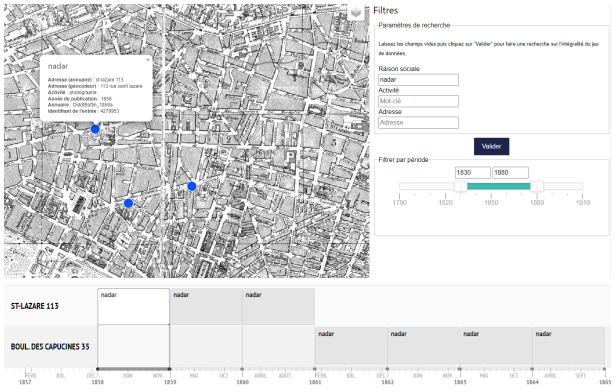


FIGURE 2 – Déménagement du photographe Nadar

FIGURE 3 – Les ateliers dont les propriétaires se nomment "Chevalier" se concentrent dans deux quartiers.

semble des scripts relatifs à ce travail, l'ontologie et l'application de visualisation :`https://github.com/soduco/ic_2023_photographes_parisiens`

5.2 Visualisation

Pour explorer les données du graphe de façon intuitive, nous avons développé une application de visualisation cartographique et temporelle. Elle permet de filtrer les données par nom de commerce, par adresse, par activité et par intervales temporels et facilite l'identification d'éventuelles corrélations spatiales et temporelles et la réponse à certaines questions de compétence. Ainsi, en figure 2, la frise temporelle associée au photographe Nadar permet de constater que son atelier a déménagé, en 1860, de la rue Saint-Lazare au 113 boulevard des Capucines. La figure 3 montre que les ateliers des frères de la famille Chevalier sont concentrés quai de l'Horloge ; seul l'un des neveux, Charles, déménage en 1831 au Palais Royal. Enfin, la figure 4 montre qu'au moins 4 photographes se sont succédés au 59 rue de Rivoli au cours de la seconde moitié du XIX^{e} siècle.

6 Conclusion et perspectives

Dans cet article, nous avons proposé une approche pour créer et analyser un graphe de connaissances géohistoriques sur des commerces d'un type donné, à partir d'annuaires du commerce anciens. Les deux stratégies de liage adoptées permettent de créer suffisament de liens entre entrées représentant un même commerce au cours du temps pour

FIGURE 4 – Phénomène de transmission probable d'un atelier entre photographes.

suivre l'évolution des entrées issues d'éditions successives. Le géocodage des entrées et leur visualisation cartographique permettent en outre d'identifier aisément des phénomènes spatiaux. Trois perspectives à cours terme sont prévues : l'explicitation des relations spatio-temporelles entre entrées, la publication du graphe sur le Web et la mise en oeuvre de l'approche pour d'autres types de commerces.

Remerciements

Ce travail a été soutenu financièrement par l'Agence Nationale de la Recherche dans le cadre du projet SODUCO (ANR-18-CE38-0013) et par le Ministère des Armées – Agence de l'innovation de défense.

Références

[1] N. Abadie, E. Carlinet, J. Chazalon, and B. Duménieu. A Benchmark of Named Entity Recognition Approaches in Historical Documents Application to 19th Century French Directories. In S. Uchida, E. Barney, and V. Eglin, editors, *Document Analysis Systems. DAS 2022.*, number 13237 in Document Analysis Systems. DAS 2022., La Rochelle, France, May 2022. Springer, Cham.

[2] Nacira Abbas, Jérôme David, and Amedeo Napoli. Linkex : A Tool for Link Key Discovery Based on Pattern Structures. In *ICFCA 2019 - workshop on Applications and tools of formal concept analysis*, Proc. ICFCA workshop on Applications and tools of formal concept analysis, pages 33–38, Frankfurt, Germany, June 2019. abbas2019a.

[3] Thilo Albers and Kalle Kappner. Perks and Pitfalls of City Directories as a Micro-Geographic Data Source. *Rationality and Competition*, Discussion Paper No. 315, January 2022.

[4] Samuel Bell, Thomas Marlow, Kai Wombacher, Anina Hitt, Neev Parikh, Andras Zsom, and Scott Frickel. Automated data extraction from historical city directories : The rise and fall of mid-century gas stations in Providence, RI. *PLOS ONE*, 15(8) :e0220219, August 2020. Publisher : Public Library of Science.

[5] Camille Bernard, Marlène Villanova-Oliver, and Jérôme Gensel. Theseus : A framework for managing knowledge graphs about geographical divisions and their evolution. *Transactions in GIS*, 2022.

[6] Duménieu Bertrand. *Un système d'information géographique pour le suivi d'objets historiques urbains à travers l'espace et le temps*. PhD thesis, Ecole des Hautes Etudes en Sciences Sociales, 2015.

[7] Galal M Binmakhashen and Sabri A Mahmoud. Document layout analysis : a comprehensive survey. *ACM Computing Surveys (CSUR)*, 52(6) :1–36, 2019.

[8] Thomas M Breuel. The OCRopus open source OCR system. In *Document Recognition and Retrieval XV*, volume 6815, page 68150F. Int. Soc. for Optics and Photonics, 2008.

[9] Marc Durand. *De l'image fixe à l'image animée : 1820-1910. Tome 2 : actes des notaires de Paris pour servir à l 'histoire des photographes et de la photographie*. Number 2. Archives nationales, Pierrefitte-sur-Seine, 2015.

[10] Del Mondo Géraldine. *Un modèle de graphe spatio-temporel pour représenter l'évolution d'entités géographiques*. PhD thesis, Université de Brest, 2011.

[11] Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. Recursive xy cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955. IEEE, 1995.

[12] Robert Isele, Anja Jentzsch, and Christian Bizer. Efficient multidimensional blocking for link discovery without losing recall. In *WebDB*, 2011.

[13] Benjamin Kiessling. Kraken-an universal text recognizer for the humanities. In *Alliance of Digital Humanities Organizations (ADHO), Éd., Actes de la conférence Digital Humanities*, Utrecht, The Netherlands, 2019.

[14] Jan Kohút and Michal Hradiš. TS-Net : OCR trained to switch between text transcription styles. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, *Document Analysis and Recognition – ICDAR 2021*, pages 478–493. Springer Int. Publishing, 2021.

[15] Aurélie Leborgne, Adrien Meyer, Henri Giraud, Florence Le Ber, and Stella Marc-Zwecker. Un graphe spatio-temporel pour modéliser l'évolution de parcelles agricoles. In *Conférence internationale francphone en analyse spatiale et géomatique SAGEO*, 2019.

[16] Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. StructuralLM : Structural pre-training for form understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, pages 6309–6318, Online, August 2021. Association for Computational Linguistics.

[17] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1) :50–70, 2020.

[18] Denis Maurel, Nathalie Friburger, Jean-Yves Antoine, Iriss Eshkol-Taravella, and Damien Nouvel. Casen : a transducer cascade to recognize french named entities. *TAL*, 52(1) :69–96, 2011.

[19] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1) :3—26, 2007.

[20] George Nagy and Sharad C Seth. Hierarchical representation of optically scanned documents. In *" International conference on Pattern Recognition*, 1984.

[21] Axel-Cyrille Ngonga Ngomo. Orchid–reduction-ratio-optimal computation of geo-spatial distances for link discovery. In *International Semantic Web Conference*, pages 395–410. Springer, 2013.

[22] Lawrence O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 15(11) :1162–1173, 1993.

[23] François Scharffe, Alfio Ferrara, and Andriy Nikolov. Data linking for the semantic web. *International Journal on Semantic Web and Information Systems*, 7(3) :46–76, 2011.

[24] Willington Siabato, Christophe Claramunt, Sergio Ilarri, and Miguel Ángel Manso-Callejo. A survey of modelling trends in temporal gis. *ACM Computing Surveys (CSUR)*, 51(2) :1–41, 2018.

[25] Ray Smith. An overview of the tesseract OCR engine. In *Int. Conf. on Doc. Analysis and Recognition*, volume 2, pages 629–633. IEEE, 2007.

[26] Phaisarn Sutheebanjard and Wichian Premchaiswadi. A modified recursive xy cut algorithm for solving block ordering problems. In *2010 2nd International Conference on Computer Engineering and Technology*, volume 3, pages V3–307. IEEE, 2010.

[27] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. SAKey : Scalable Almost Key Discovery in RDF Data. In Springer Verlag, editor, *In proceedings of the 13th International Semantic Web Conference, ISWC 2014*, volume Lecture Notes in Computer Science of *The Semantic Web – ISWC 2014*, pages 33–49, Riva del Garda, Italy, October 2014. Editions Springer.

[28] Christoph Wick, Christian Reul, and Frank Puppe. Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition. *Digital Humanities Quarterly*, 14(1), 2020.

[29] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm : Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

Construction d'un graphe de connaissance à partir des annotations manuelles de textes de zoologie antique

A. Barbe[1], M. Tounsi Dhouib[2], C. Faron[2], M. Corneli[1], A. Zucker[1]

[1] Université côte d'Azur, CEPAM, CNRS, France
[2] Université côte d'Azur, INRIA, CNRS, I3S, France

arnaud.barbe@univ-cotedazur.fr, dhouib@i3s.unice.fr, faron@i3s.unice.fr,
marco.corneli@univ-cotedazur.fr, Arnaud.zucker@univ-cotedazur.fr

Résumé

Ce travail est réalisé dans le cadre de l'IRN Zoomathia qui vise l'étude de la transmission des savoirs zoologiques de l'Antiquité au Moyen Âge. Dans ce contexte, un premier travail d'annotation manuelle de l'"Histoire Naturelle" de Pline l'Ancien sur la zoologie antique (livres 8-11) en utilisant des concepts rassemblés dans le thésaurus TheZoo a été réalisé par des spécialistes de l'Antiquité. Cependant, ces annotations ont été réalisées avec des commentaires Word, ce qui rend complexe l'exploitation de ces connaissances par les chercheurs épistémologues, historiens et philologues dans leur travail d'analyse de ces textes anciens. Dans cet article, nous présentons notre approche de transformation de ces annotations manuelles en graphe de connaissance RDF permettant l'intégration et l'interrogation des connaissances extraites dans le but d'aider les chercheurs dans leur travail d'analyse de ces textes et de la transmission des connaissances à travers eux. Afin de valider la pertinence du modèle et le graphe de connaissance, nous avons recueilli auprès d'un expert du domaine un ensemble de questions de compétences que nous avons traduites en SPARQL pour y répondre en interrogeant le graphe de connaissance produit.

Mots-clés

Ontologie, Annotation sémantique de textes latins, Graphes de connaissances, Données liées et vocabulaires, Histoire de la zoologie.

Abstract

This work is carried out in the framework of the IRN Zoomathia which aims to study the transmission of zoological knowledge from Antiquity to the Middle Ages. In this context, a first work of manual annotation of Pline's Naturalis Historia (books 8-11) on ancient zoology using concepts gathered in the thesaurus TheZoo was done by classicists. However, these annotations have been stored as comments in Word documents, which complicates the exploitation of this knowledge by epistemology, history and philiology researchers in their analysis of these ancient texts. In this article, we present our approach to transform manual annotations into an RDF knowledge graph allowing the integration and the interrogation of relevant knowledge in order to support researchers in their analysis of these texts and knowledge transmission through them. In order to validate the relevance of the model as well as the knowledge graph, we elicited from a domain expert a set of competency questions that we translated in SPARQL to answer them by querying the knowledge graph.

Keywords

Semantic Annotation of Latin Texts, Knowledge Graphs, Ontologies, Linked Data and Vocabularies, History of Zoology.

1 Introduction

Les historiens et les philologues doivent faire face quotidiennement à une quantité énorme de ressources textuelles. Malgré les efforts de numérisation, les outils proposés ne répondent pas aux exigences épistémologiques en ne permettant souvent que des recherches lexicales et quantitatives des données. Les chercheurs expriment un besoin d'outils plus intelligents afin de réaliser des recherches plus élaborées qui nécessitent une annotation sémantique plus riche. Le Réseau de Recherche International (IRN) Zoomathia [1] vise l'étude de la constitution et de la transmission des connaissances zoologiques de l'Antiquité au Moyen Âge, à travers des ressources variées, et considère en particulier l'information textuelle. Dans ce contexte, un premier travail d'annotation manuelle de quatre chapitres de l'"Histoire Naturelle" de Pline a été réalisé par un chercheur en littérature latine. Ainsi, sous la forme de commentaires dans un document Word, chaque texte latin a été annoté avec les concepts du thésaurus TheZoo [2]. Malgré l'énorme effort réalisé et le temps passé à annoter ces textes, ces annotations restent inexploitables en termes de formalisation du savoir et d'intégration de ces connaissances avec d'autres sources de connaissances. L'objectif de notre travail est de transformer ces annotations manuelles en graphe de connaissance permettant ainsi l'intégration et l'interrogation des connaissances extraites dans le but de proposer des possibilités de recherche automatique plus riches et répon-

1. https ://www.cepam.cnrs.fr/sites/zoomathia/
2. https ://opentheso.huma-num.fr/opentheso/ ?idt=th310

dant mieux aux besoins des chercheurs qui étudient cette littérature scientifique. Nous avons identifié trois questions de recherche : (i) Quels types de connaissances devons-nous représenter afin d'aider les chercheurs dans leur travail d'analyse et de transmission de savoir zoologique ? (ii) Quelles ontologies existantes pouvons-nous réutiliser pour représenter ces documents ? (iii) Quelle approche pouvons-nous définir pour réutiliser les annotations manuelles faites par les linguistes et les rendre exploitables ?
Notre approche de construction du graphe de connaissance repose sur (i) la proposition d'un modèle qui réutilise des ontologies et vocabulaires existants afin de structurer et représenter les annotations manuelles des textes de zoologie ancienne, (ii) l'explicitation de questions de compétences auprès d'historiens et philologues intéressés par la transmission des connaissances zoologiques. Le processus de construction du graphe de connaissances comprend cinq étapes successives : (i) la reconnaissance des entités pertinentes dans les annotations manuelles, (ii) le liage de ces entités avec les concepts du thésaurus TheZoo, (iii) l'extraction des contenus textuels des chapitres et paragraphes du texte annoté, (iv) le liage des paragraphes avec les annotations, et enfin (v) la génération du graphe RDF capturant à la fois le contenu textuel et la structure de l'Histoire Naturelle de Pline et les annotations du texte à l'aide de l'outil morph-xr2rml [5].
Cet article est organisé comme suit. Dans la section 2, nous présentons une synthèse des approches de construction de graphe de connaissance à partir de textes anciens (médiévaux) ainsi que les vocabulaires réutilisés dans ce travail. Dans la section 3, nous présentons un ensemble de questions de compétences représentatives des besoins des experts en termes d'exploitation des annotations générées. La section 4 décrit le modèle sémantique du graphe de connaissance. Dans la section 5, nous détaillons le processus que nous avons utilisé pour la génération de ce graphe de connaissance. Enfin, dans la section 6 nous présentons des requêtes SPARQL qui implémentent des questions de compétences élicitées et dont la réponse peut être recherchée dans le graphe de connaissance produit, validant ainsi celui-ci.

2 État de l'art

2.1 Construction de graphes de connaissance à partir de textes anciens

Plusieurs travaux dans la littérature ont traité la problématique d'analyse et de structuration des ressources culturelles et historiques de l'Antiquité au Moyen Âge. Des premiers travaux de recherche français s'inscrivent dans les projets SourceEncyMe4 [3] et Ichtya5 [4] portant sur la structuration d'encyclopédies médiévales en XML selon le modèle TEI et l'annotation manuelle de ces sources de données.
D'autres travaux ont fait appel aux modèles du web sémantique afin d'annoter sémantiquement des collections du patrimoine culturel et faciliter la recherche sémantique au sein de celles-ci. Le travail présenté dans [7] combine des techniques du web sémantique et du traitement automatique du langage naturel afin d'extraire automatiquement des informations à partir de textes de zoologie antique. Un modèle de publication collaboratif pour les données culturelles a été présenté dans [2]. Ce travail présente aussi des principes de conception pour la création de portails sémantiques destiné à la recherche et aux applications en Humanités Numériques. Une plate-forme orientée ontologie a été présentée dans [1] dont le but est d'aider les utilisateurs à identifier et à caractériser de nouvelles entités pour annoter les archives historiques en utilisant des techniques d'extraction automatique d'informations et les informations récupérées dans des ensembles de données externes dans le *Linked Open Data*.

2.2 Vocabulaires et ontologies existantes

Pour représenter à la fois le corpus littéraire et les annotations sémantiques extraites de ce corpus, nous avons ré-utilisé un ensemble de vocabulaires et d'ontologies. Nous avons tout d'abord utilisé le vocabulaire schema.org [5] afin de représenter la structure des textes (c.-à-d. chapitres, paragraphes, auteur, éditeur...). Ce vocabulaire propose un ensemble de classes et propriétés génériques visant à décrire initialement des ressources du web. Nous avons choisi d'utiliser ce vocabulaire car il nous permettra à terme d'intégrer facilement d'autres types de ressources tels que des images et des vidéos. Nous avons aussi utilisé le vocabulaire Web Annotation Vocabulary (OA) [6] qui est une recommandation W3C pour représenter les zones textuelles des annotations manuelles. Ce vocabulaire permet de représenter de manière uniforme des annotations sur le Web dans un format interopérable [3]. Finalement, nous avons utilisé le vocabulaire de domaine TheZoo [4] afin de lier les entités extraites des annotations sémantiques aux concepts du thésaurus. Ce vocabulaire est conçu pour représenter et structurer hiérarchiquement tous les termes d'intérêt pour l'étude de l'histoire de la zoologie antique et médiévale à partir de trois types de corpus : (i) Textuel, (ii) Iconographique et (iii) Archéologique. TheZoo contient 6019 concepts structurés en 11 niveaux hiérarchiques. Ces concepts concernent différents aspects de la description d'animaux comme, par exemple, le concept d'anatomie interne (*internal anatomy*), les noms d'animaux (*tiger*) et de lieux géographiques (*Geographic space*). Une hiérarchie permet de classifier avec précision les concepts comme par exemple le concept de *tigre* dans la hiérarchie de la famille des organismes vertébrés : "*eumetazoa* > *bilateria* > *deuterostomia* > *vertebrata* > *tetrapoda* > *mammalia* > *carnivora* > *feliformidae* > *felidae* > *pantherinae* > *tigre*". Les concepts sont également regroupés en 14 collections qui font office de méta-concept qui leur offre un sens supplémentaire, comme la collection des *Anthroponymes* rassemblant les différents noms de personnes et d'animaux nommés par des humains ou la collection des *Archéotaxons* qui rassemble les taxons d'animaux antiques.

3. http ://sourcencyme.irht.cnrs.fr
4. http ://www.unicaen.fr/recherche/mrsh/document_numerique/projets/ichtya
5. https://schema.org/docs/about.html

3 Questions de compétences

Afin de déterminer la spécificité des connaissances à représenter, nous avons collecté et explicité sept questions de compétences (QC) formulées par les experts dans le but de comprendre précisément leurs attentes et les besoins des chercheurs du domaine afin d'apporter à ces derniers une réponse adéquate en terme d'exploration des liens entre les concepts du domaine et leur contexte de co-occurence dans les textes étudiés. Nous présentons ici des exemples de ces QC.

QC1. Quels sont les animaux qui construisent un habitat ? Le besoin des chercheurs est d'identifier dans la littérature les animaux capables de construire un habitat favorable et adapté à leurs besoins.

QC2. Quelles anecdotes mettent en relation un homme et un animal ? Le besoin des chercheurs est d'identifier les passages textuels qui permettent de repérer des interactions entre l'humain et l'animal, en particulier des formes de complicité ou de coopération et des formes d'hostilité ou de prédation.

QC3. Quels sont les remèdes (thérapeutiques) dont un ingrédient est une partie d'animal, e.g. la langue (ou un morceau de langue) ? Cette question permet aux chercheurs d'identifier l'ensemble des animaux qui ont été utilisés pour des raisons médicales et plus précisément une partie exploitée du corps de l'animal.

QC4. Quels sont les animaux qui communiquent entre eux ? Le besoin des chercheurs est d'identifier le texte où il est question d'un type de communication inter-individuelle dans une espèce animale.

QC5. Quels sont les animaux capables de jeûner et quelles sont les informations sur la fréquence ou le rythme des repas ? Cette question permet de discriminer des pratiques alimentaires et de mesurer la pertinence des savoirs antiques sur ce point.

QC6. Quelles sont les données transmises sur le temps de gestation des animaux ? Le besoin des chercheurs est d'identifier les passages textuelles qui permettent de récupérer des informations sur le temps de gestation des animaux.

4 Modèle proposé

4.1 Représentation de la structure et du contenu de l'Histoire Naturelle de Pline

Pour représenter et décrire les textes annotés, nous avons utilisé le vocabulaire *Schema* pour capturer la sémantique de la décomposition de l'oeuvre de Pline en chapitres et paragraphes. Ainsi, l'Histoire Naturelle de Pline est représentée par une instance de la classe `schema:Book` dont l'auteur est décrit par la propriété `schema:author`, le titre via `schema:headline` et l'édition via `schema:editor`. Un chapitre est une instance de la classe `schema:Chapter`, il est relié à une oeuvre via la propriété `schema:isPartOf` et le numéro du chapitre est décrit via la propriété `rdf:value`.

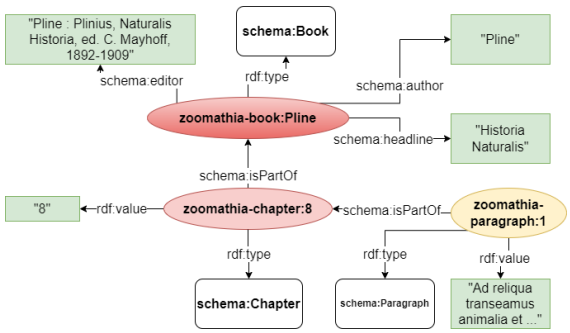


FIGURE 1 – Exemple de graphe RDF représentant un paragraphe du chapitre 8 de l'Histoire Naturelle de Pline.

Enfin, le lien entre les paragraphes et leur chapitre est représenté par la propriété `schema:isPartOf` et le contenu textuel du paragraphe est capturé comme valeur de la propriété `rdf:value`. La figure 1 présente un exemple de graphe RDF représentant le premier paragraphe du chapitre 8 de l'Histoire Naturelle de Pline.

4.2 Représentation des annotations de l'Histoire Naturelle de Pline

Afin de représenter les annotations manuelles du texte de l'Histoire Naturelle de Pline, nous avons réutilisé le vocabulaire *OA*. Une annotation a_i est une indication qu'une mention m_e d'un concept c a été identifiée dans le ou les paragraphes de l'un des quatre chapitres de l'Histoire Naturelle de Pline. Une annotation a_i est représentée comme une instance de la classe `oa:Annotation` et est décrite comme suit :

- a_i est reliée avec la propriété `oa:hasBody` à un concept c dans un vocabulaire de domaine, ici le thesaurus TheZoo.
- a_i est reliée avec la propriété `oa:hasTarget` à sa cible qui elle-même est reliée avec la propriété *oa :hasSelector* la zone de texte sélectionnée pour l'annotation et avec la propriété `oa:hasSource` au paragraphe contenant cette zone de texte. Cette zone de texte est décrite par sa valeur littérale (propriété `oa:exact`) et son début et sa fin relativement au début du paragraphe source (propriétés `oa:start` et `oa:end`).

La figure 2 présente un exemple d'annotation du paragraphe 14 du chapitre 11 de l'Histoire Naturelle de Pline. Cette annotation porte sur le texte "*tigrium rapinas*" qui est accessible via la propriété `oa:exact`. Cette annotation a été mise en correspondance avec le concept idc :5066 du thesaurus TheZoo dont un label est "Tigre" et qui est un sous concept du concept "Pantherinae".

En utilisant cette représentation RDF, nous avons pu modéliser d'une part les paragraphes des chapitres qui ont été annotés manuellement par les experts et d'autre part, la mise en correspondance de ces annotations avec les concepts des ontologies et vocabulaires du domaine (ici, le thésaurus TheZoo). Cette représentation offre la possibilité aux chercheurs d'explorer non seulement les occurrences et les

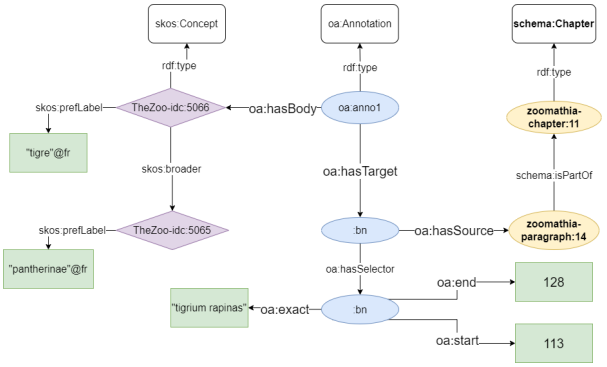


FIGURE 2 – Exemple de graphe RDF représentant l'annotation de "*tigrium rapinas*" présente dans le paragraphe 14 du chapitre 11 de l'Histoire Naturelle de Pline.

co-occurrences des annotations dans les textes mais aussi d'obtenir plus d'informations et de raisonnements sur ces annotations grâce au liage du graphe avec les ontologies et vocabulaires du domaine.

5 Construction du graphe de connaissance

5.1 Description du corpus de textes annotés

Nous avons construit un graphe de connaissances à partir des annotations manuelles du texte latin des chapitres 8 à 11 de l'Histoire Naturelle de Pline qui traitent de zoologie, respectivement des animaux terrestres, des animaux marins, des oiseaux et des insectes. Ces livres totalisent 911 paragraphes. Ces paragraphes ont été manuellement annotés par des linguistes avec les concepts du thésaurus TheZoo.

Ces annotations manuelles ont une granularité variable (un mot, un groupe de mots, un ou plusieurs paragraphes) afin de délimiter le contexte du concept annotant le texte. Le système de commentaire de Word permet de définir ces zones d'annotation et le texte de ces commentaires fait référence au(x) concept(s) du thésaurus en fonction des motifs suivants :

- "*concept*" : référence directe à un concept
- "*concept1 : concept2 : ...*" : référence à une hiérarchie de concepts où concept1 est parent de concept2
- "*concept1 ; concept2 ; ...*" : référence à des concepts distincts annotant la même portion de texte
- "*collection : concept*" : référence à un concept faisant partie d'une collection
- "*concept1 : concept2, concept3, ...*" : référence à des concepts des descendants directs d'un autre
- combinaisons des motifs précédents.

Ainsi, notre corpus de 4 livres contient 7,283 commentaires à partir desquels 13,241 références de concepts du thésaurus TheZoo ont été annotés.

5.2 Processus de lifting

La figure 3 présente le processus de transformation des annotations manuelles du texte de l'Histoire Naturelle de

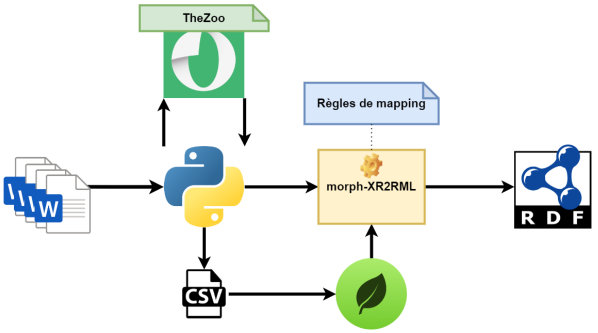


FIGURE 3 – Schéma général du processus de construction du graphe de connaissance

Pline et la construction du graphe RDF. La première étape de transformation consiste à extraire les annotations manuelles à partir des commentaires des fichiers Word. Ces informations sont stockées dans des fichiers xml internes au document word. Les informations concernant (i) le chapitre, (ii) le paragraphe, (iii) la portion du texte en latin qui a été sélectionné et qui correspond à la mention et enfin (iv) le texte du commentaire de l'expert qui correspond aux labels des concepts de TheZoo sont renseignées. La deuxième étape consiste à utiliser l'annotation manuelle des concepts au sein de requêtes SPARQL pour rechercher dans le thésaurus l'URI du concept extrait grâce à son label. Un exemple de requête que nous avons utilisée est présenté dans le listing 1. Cette requête permet de faire la correspondance entre le label extrait et le concept du thésaurus en conservant la hiérarchie de concept ou la collection. A cause de la présence de l'homonymie dans le thésaurus, et afin d'éliminer tout cas d'ambiguïté dans l'étape de la recherche du concept, nous avons choisi de vérifier l'appartenance du concept à une collection ou à une branche de la hiérarchie.

```
SELECT ?x WHERE {
{?x a ?type; skos:prefLabel ?label.
?y skos:prefLabel ?collection; skos:member ?x.
FILTER(lang(?label) = "en").
FILTER("{label}" in (ucase(str(?label)),
  lcase(str(?label)), str(?label))).
FILTER( "{parent}" in (ucase(str(?collection)),
  lcase(str(?collection)), str(?collection))).}
UNION { ?x a ?type; skos:prefLabel ?label;
       skos:broader+ ?y.
       ?y skos:prefLabel ?concept;
FILTER(lang(?label) = "en").
FILTER("{label}" in (ucase(str(?label)),
 lcase(str(?label)), str(?label))).
FILTER( "{parent}" in (ucase(str(?concept)),
  lcase(str(?concept)), str(?concept))).}}
```

Listing 1 – Requête SPARQL de recherche de concept dans le thésaurus

Toutes ces informations (les paragraphes de commentaire, les concepts extraits, etc.) sont finalement enregistrées dans un fichier CSV qui va être injecté dans le système de gestion de base de données orienté documents MongoDB.

Puis, nous avons utilisé l'outil morph-xR2rml afin de transformer les annotations produites en un graphe RDF. Pour

Nbre de paragraphes	911
Nbre de commentaires	7283
Nbre d'entités reconnues	13241
Nbre d'entités liées	11590
Nbre d'entités non liées	2632
Nbre de triplet RDF générés	88184

TABLE 1 – Caractéristiques du graphe de connaissances produit

cela, nous avons écrit un ensemble de règles de mapping génériques permettant de générer nos triplets RDF. Les règles de mapping sont écrites en RDF à l'aide du vocabulaire xr2rml [6] basé sur r2rml [7] qui expriment des patrons de transformation de données provenant d'une base de données. Une règle de transformation définit une ressource de type *rr :TripleMap* qui est décrite par un unique sujet *rr :subjectMap*, une source logique *rr :logicalSource* qui représente la base de données MongoDB [8] et un ensemble de propriétés *rr :predicateObjectMap*. Le listing 2 présente un exemple de règle de mapping qui permet de générer une partie d'une annotation avec "rr" et "xrr" les préfixes des ontologies "r2rml" et "extended r2rml".

```
<#Anno>
a rr:TripleMap;
xrr:logicalSource [
    xrr:query """db.Annotation.find()""";];
rr:subjectMap [
        rr:template
"http://www.zoomathia.com/annotation/
sha1({$.id}_{$.chapter}_{$.paragraph})";
        rr:class oa:Annotation;];
rr:predicateObjectMap [
        rr:predicate oa:hasBody;
        rr:objectMap [
        rr:template
"https://opentheso.huma-num.fr/
?idc={$.concept}&idt=th310";];];
rr:predicateObjectMap [
    rr:predicate oa:hasTarget;
        rr:objectMap [
                rr:template "TargetBN{$._id}";
                rr:termType rr:BlankNode;];].
```

Listing 2 – Extrait d'une règle de mapping xR2RML

Nous avons défini deux bases de règles de mapping : (i) l'une pour décrire la structuration des textes de Pline en livres, chapitres et paragraphes, (ii) l'autre pour décrire les annotations de ces textes en les liant avec les paragraphes annotés, le texte de l'annotation et le lien vers les concepts de TheZoo.

A la fin de ce processus, nous avons pu extraire automatiquement 11590 concepts à partir des annotations manuelles des experts, et nous avons généré 88184 triplets RDF. Le tableau 1 résume les caractéristiques du graphe de connaissance produit.

En utilisant cette approche, nous avons échoué à lier 2632 entités annotées manuellement par les experts à des concepts de TheZoo. Cela s'explique par des irrégularités dans certaines annotations manuelles. En effet, comme la tâche d'annotation manuelle est une tâche fastidieuse pour l'annotateur, nous avons été confrontés à des problèmes d'irrégularités des règles d'annotation manuelle (faute de frappe, utilisation du pluriels, ...). Dans notre processus de transformation, nous utilisons le texte de ces annotations manuelles dans le filtre des requêtes SPARQL pour rechercher des correspondances avec les labels de concepts de TheZoo. Des annotations non uniformes engendrent des problèmes de correspondance. Par exemple, pour annoter les informations concernant la taille des animaux, l'annotateur utilise en général la syntaxe "`size : relative size`" qui fait référence au concept "relative size" dans le thésaurus. Dans certains cas, l'annotateur peut se contenter de mentionner le terme "relative" en utilisant cette syntaxe "`size : relative`".

6 Evaluation de la qualité du graphe produit

6.1 Evaluation du processus d'extraction de connaissances

Nous pouvons évaluer la qualité du graphe produit en terme de la qualité du processus d'extraction des connaissances à partir des annotations de texte en utilisant les métriques classiques de précision et rappel. Notre approche est conçue de telle manière que la précision est maximale (P=1). Notre processus de lifting des annotations en RDF a généré 11590 liens vers les concepts de TheZoo et 2632 annotations n'ont pu être liées (erreurs d'orthographe, typographiques, etc. dans les annotations, concepts absents ou labels manquants dans le thesaurus). Ainsi, la performance de notre processus en terme de rappel est de 0.814. L'analyse des annotations qui n'ont pu être liées au thesaurus avec les experts du domaine va nous permettre d'améliorer la qualité des annotations et du thesaurus.

6.2 Implémentation et réponse aux questions de compétence recueillies

Nous avons utilisé les questions de compétences présentées en section 3 pour valider le graphe RDF produit. A travers les questions de compétences traduites en SPARQL, nous avons vérifié que le graphe produit permet de répondre aux besoins des experts en terme d'exploration des connaissances zoologiques. Toutes les questions de compétences élicitées ont été formalisées en SPARQL [9] et validées par les experts du domaine. Nous ne présentons ici que deux de ces requêtes avec leurs formalisations et les différents résultats avec le retour de l'expert.

QC1. Quels sont les animaux qui construisent un habitat ? L'intention du chercheur derrière cette question est d'identifier les paragraphes des chapitres où l'auteur mentionne des animaux capables de construire leur habitat pour les étudier ensemble. Le listing 3 présente la requête SPARQL qui implémente QC1.

6. https ://www.i3s.unice.fr/~fmichel/xr2rml_specification_v5.html
7. https ://www.w3.org/TR/r2rml/
8. https ://www.mongodb.com
9. https://github.com/Wimmics/zoomathia/tree/main/Pline

```
SELECT DISTINCT ?paragraph ?name_animal
?name_construction
WHERE {
?annotation1 a oa:Annotation;
               oa:hasBody ?animal;
               oa:hasTarget [oa:hasSource
               ?paragraph; oa:hasSelector
               [oa:exact?mention_animal]].
?annotation2 oa:hasBody ?construction;
             oa:hasTarget [oa:hasSource
             ?paragraph; oa:hasSelector
             [oa:exact ?mention_construction]].
?animal  a skos:Concept;
         skos:prefLabel ?name_animal.
<https://opentheso.huma-num.fr/idg=MT_10
&idt=th310> skos:member ?animal.
?construction skos:prefLabel ?name_construction;
              skos:broader+ <https://opentheso.
              huma-num.fr/?idc=105466&idt=th310>.
FILTER (lang(?name_animal) = "en").
FILTER (lang(?name_construction) = "en")
}ORDER BY ?paragraph
```

Listing 3 – Requête SPARQL de la CQ1

Le résultat de cette requête indique, par exemple, que le paragraphe *104* du livre *10* mentionne que les Méropidae ("*bee eater*") construisent des nids ("*nest*").
QC6. Quelles sont les données transmises sur le temps de gestation des animaux ? L'intention du chercheur derrière cette question est d'identifier les paragraphes des chapitres qui mentionnent des informations du temps de gestation des animaux. Le listing 4 présente la requête SPARQL qui implémente QC6.

```
SELECT DISTINCT ?paragraph ?name_animal
?mention_animal ?mention_pregnancy
WHERE {?annotation1 a oa:Annotation;
               oa:hasBody ?animal;
               oa:hasTarget [
               oa:hasSource ?paragraph;
                oa:hasSelector [oa:exact
            ?mention_animal]].
?animal a skos:Concept;
        skos:prefLabel ?name_animal.
<https://opentheso.huma-num.fr/?idg=MT_10&idt
=th310> skos:member ?animal.
?annotation2 oa:hasBody <https://opentheso.huma-
num.fr/?idc=105364&idt=th310>;
             oa:hasTarget [
             oa:hasSource ?paragraph;
             oa:hasSelector [oa:exact
         ?mention_pregnancy]].
FILTER (lang(?name_animal) = "en").
}ORDER BY ?paragraph
```

Listing 4 – Requête SPARQL de la CQ6

7 Conclusion et travaux futurs

La capitalisation des connaissances et le développement de meilleures techniques de recherche d'informations est devenue une tâche cruciale dans la communauté des humanités numériques pour les chercheurs soucieux de valoriser le patrimoine culturel. Dans cet article, nous avons présenté un graphe de connaissance que nous avons construit à partir des annotations manuelles par des experts de l'oeuvre de Pline en utilisant le thésaurus TheZoo. Dans le graphe RDF produit, nous avons pu : (i) capturer le contexte d'apparition des différentes annotations, (ii) les décrire d'une manière structurée grâce aux vocabulaires standards du web sémantique, et (iii) lier ces annotations manuelles avec le vocabulaire de domaine TheZoo. Le graphe produit permet une interrogation uniforme, avancée à l'aide de requêtes SPARQL et qui exploite les contextes d'apparition et les liens entre les concepts du vocabulaire du domaine. La génération de ce graphe de connaissance a également permis d'identifier des problèmes d'irrégularité d'annotation. Nous avons partiellement contourné ce problème avec une recherche de correspondance approximative entre les entités extraites et les concepts du thésaurus TheZoo, par exemple en recherchant des inclusions plutôt que des égalités de chaînes de caractères entre l'entité extraite et les labels des concepts du thésaurus. Cependant, cette approche a des limites, car elle génère du bruit : par exemple, "relative" est contenue dans "relative size" mais aussi dans "tail relative size".Les entités non liées à TheZoo ont ainsi fait apparaître le besoin de corriger certaines annotations et/ou réviser ou enrichir le thésaurus TheZoo. Ce travail est prévu prochainement dans le cadre du projet Zoomathia. Une perspective de ce travail, est d'automatiser la tâche d'annotation des textes, fastidieuse pour les experts et source d'erreur. Le graphe RDF produit constitue des données de très bonne qualité pour l'entraînement d'algorithmes d'apprentissage sur lesquels reposeront l'approche que nous souhaitons développer. Une autre perspective est de faciliter l'exploitation de ce graphe RDF par les experts du domaine, philologues et historiens, qui ne sont pas spécialistes des modèles du web sémantique, en développant des interfaces de visualisation plus intelligibles et intuitives.

Références

[1] Davide Colla, Annamaria Goy, Marco Leontino, Diego Magro, and Claudia Picardi. Bringing semantics into historical archives with computer-aided rich metadata generation. *Journal on Computing and Cultural Heritage (JOCCH)*, 15(3) :1–24, 2022.

[2] Eero Hyvönen. Digital humanities on the semantic web : Sampo model and portal series. *Semantic Web*, (Preprint) :1–16, 2022.

[3] Jin-Dong Kima, Karin Verspoorb, Michel Dumontierc, and K Bretonnel Cohend. Semantic representation of annotation involving texts and linked data resources. *Semantic Web journal*, 2015.

[4] Irene Pajón Leyra, Arnaud Zucker, and Catherine Faron Zucker. Thezoo : un thesaurus de zoologie ancienne et médiévale pour l'annotation de sources de données hétérogènes. *Archivum Latinitatis Medii Aevi*, 73 :321–342, 2015.

[5] Franck Michel, Loïc Djimenou, Catherine Faron Zucker, and Johan Montagnat. Translation of Relational and Non-Relational Databases into RDF with xR2RML. In *11th International Confenrence on Web Information Systems and Technologies (WEBIST'15)*, Proceedings of the WebIST'15 Conference, pages 443–454, Lisbon, Portugal, October 2015.

[6] Robert Sanderson, Paolo Ciccarese, and Benjamin Young. Web annotation ontology. https://www.w3.org/TR/annotation-vocab/, 2017.

[7] Molka Tounsi, Catherine Faron Zucker, Arnaud Zucker, Serena Villata, and Elena Cabrio. Studying the history of pre-modern zoology with linked data and vocabularies. In *The First International Workshop Semantic Web for Scientific Heritage at the 12th ESWC 2015 Conference*, 2015.

Graphes de Connaissances et Ontologie pour la Representation de Données Immobilières Issues d'Annonces en Texte Libre

Lucie Cadorel[1,2,3], Andrea G. B. Tettamanzi[1,2], Fabien Gandon[1,2]

[1] Inria Sophia-Antipolis
[2] I3S, Université Nice Côte d'Azur, CNRS
[3] Septeo Proptech

lucie.cadorel@inria.fr, andrea.tettamanzi@univ-cotedazur.fr, fabien.gandon@inria.fr

Résumé

Nous décrivons nos premiers travaux sur la conception d'une ontologie et d'un graphe de connaissances pour la représentation des données immobilières issues d'annonces. Nous identifions plusieurs scénarios motivants à partir des données extraites afin de justifier notre modélisation. Puis nous proposons une modélisation pour représenter les données spatiales incertaines et leurs géométries dans un graphe de connaissances.

Mots-clés

Ontologie, Graphe de Connaissances, Texte, Immobilier, Données spatiales

Abstract

We describe our initial work on the design of an ontology and a knowledge graph for the representation of real estate data from advertisements. We identify several motivating scenarios from the extracted data to justify our modelling. Then we propose a model to represent uncertain spatial data and their geometries in a knowledge graph.

Keywords

Ontology, Knowlede Graph, Text, Real Estate domain, Spatial data

1 Introduction et Motivations

1.1 Contexte

Le secteur de l'immobilier joue un rôle important dans l'économie, au point que son poids dépasse celui de l'industrie et de l'agriculture réunies [2]. Les données immobilières sont de ce fait très étudiées par les professionnels du secteur, notamment pour estimer les prix et les tendances du marché. Ces données sont à la fois temporelles et spatiales, et proviennent de différentes sources (notaires, agents immobiliers, particulier, etc.). Plus particulièrement, les annonces immobilières constituent une source très abondante de données facilement accessibles, exhaustives et mises à jour. Les caractéristiques du bien et de son environnement sont en général décrites par l'annonceur, et peuvent être extraites à l'aide d'un modèle de langage [3, 4]. Cependant, ces données ne sont pas structurées, ce qui constitue un obstacle à leur exploitation, notamment pour effectuer du raisonnement. Si les ontologies et les graphes de connaissances peuvent aider à modéliser et représenter les informations issues des textes d'annonces, et faciliter leur interopérabilité, il en reste que certaines informations, notamment la description de l'environnement et de la localisation, sont parfois incertaines et floues (par exemple "proche du centre-ville"), et nécessitent une représentation appropriée.

1.2 Motivations et Questions de compétences

Afin de spécifier notre formalisation, l'ontologie attenante, et les applications futures, nous avons défini plusieurs cas d'usage. Pour cela, nous avons suivi la méthodologie "classique" des scénarios motivants et questions de compétences [1]. Nous avons d'abord identifié et dialogué avec des utilisateurs potentiels, notamment des professionnels de l'immobilier et des chercheurs en géographie. A partir de ce recueil, nous avons identifié des scénarios ainsi que les questions de compétences qui en découlent. Les tableaux 1, 2, 3 et 4 nomment (*Nom*), décrivent (*Desc.*), exemplifient (*ex 1, 2*) les scénarios et déduisent des questions de compétences (*QC*).

Nom	Recherche d'un bien immobilier
Desc.	Un acheteur recherche un bien immobilier dans une ville et un secteur particulier. Pour prendre sa décision, il a besoin de connaître les caractéristiques du bien, son prix et son environnement.
Ex. 1	Mathilde recherche un bien à acheter à Cannes, avec vue sur la mer et proche de la Croisette. Elle voudrait un 2 pièces pour moins de 300 000 euros.
Ex. 2	Paul déménage à Nice pour le travail. Il ne connaît pas la région et voudrait trouver un appartement proche des transports et des commerces, tout en étant au calme.
QC	- Quelles sont les caractéristiques du bien ? - Dans quelle ville se trouve le bien ? - Dans quel quartier se trouve le bien ? - À proximité de quels lieux se trouve le bien ? - Dans quel environnement (ex. sonore) est le bien ?

TABLE 1 – Scénario 1 : Recherche d'un bien immobilier

Nom	Étude du marché immobilier
Desc.	Un agent immobilier a besoin de comprendre le marché et les biens en vente/vendus pour positionner le bien qu'il a à vendre sur le marché.
Ex. 1	Denis a obtenu le mandat de vente d'une très belle maison à Antibes. Cependant, ce n'est pas le secteur sur lequel il a l'habitude de travailler. Il aimerait en savoir plus sur les prix de ce secteur et les biens vendus et/ou en vente similaires à ce bien avant de mettre en ligne son annonce.
Ex. 2	David est promoteur et cherche le meilleur quartier où construire son prochain immeuble. Il a besoin pour chaque quartier de savoir les prix au m^2 et les services (écoles, transports, etc.) s'y trouvant.
QC	- Quels sont les autres biens du même secteur ? - Sont-ils similaires à celui à vendre ? - Quel est le prix moyen ? - Quels sont les services souvent mentionnés dans ce secteur ? - Quels sont les autres lieux mentionnés dans ce secteur ? - Quels sont les volumes de ventes des quartiers ? - Quels sont les prix au m^2 des quartiers ?

TABLE 2 – Scénario 2 : Etude du marché immobilier

Nom	Analyse du territoire
Desc.	Les agents immobiliers ont une bonne connaissance du territoire et en parlent au travers des annonces immobilières. Ceci permet d'avoir de nouvelles connaissances (plus proche du réel) qui peuvent être analysées.
Ex. 1	Alicia est chercheuse en géographie et voudrait étudier la forme sociale des espaces urbains afin de comprendre quelle partie du territoire est plus adaptée à une population ou à une autre.
Ex. 2	Clément travaille dans le service d'urbanisation de la mairie de Nice et voudrait comprendre la politique de la ville en matière de transports (zone qui manque de transports) et d'accès aux services par la population pour ajuster ses recommandations.
QC	- Comment est perçu un quartier (recherché, résidentiel, etc.) par les agents immobiliers ? - Quels services sont mentionnés dans un secteur ? - Quels services ne sont pas mentionnés ? - Quels autres lieux sont mentionnés dans ce secteur ? - Est-ce que certains lieux sont souvent / toujours / jamais mentionnés ensemble ? - Dans quelle partie de la ville parle-t-on le plus du tramway ? - Est-ce que les annonces qui mentionnent les transports en commun sont situé loin de certains lieux centraux (centre-ville) ? - Quelles sont les zones dans lesquelles les déplacements piétons sont mentionnés ? - Est-ce qu'un lieu est inclus dans un autre lieu ?

TABLE 3 – Scénario 3 : Analyse du territoire

1.3 Contributions

Nous avons conçu une ontologie pour la représentation des données immobilières issues du texte des annonces et répondant aux scénarios et questions identifiés. Pour cela, nous proposons une représentation des données spatiales floues et incertaines dans un graphe de connaissances. Le reste de cet article est organisé de la manière suivante. Dans la section 2, nous présentons l'état de l'art et ses limites à la lumière des scénarios présentés. La section 3 détaille la théorie des ensembles flous appliquée aux objets spatiaux incertains. Enfin, nous présentons nos choix techniques pour la modélisation de l'ontologie dans la section 4.

Nom	Utilisation d'annotations textuelles
Desc.	Les entités extraites à partir du texte peuvent constituer un très grand jeu de données pour la reconnaissance d'entités nommées (géographiques). Néanmoins, il faudra ajouter un terme de confiance pour chaque entité extraite.
Ex. 1	Julien est doctorant en NLP et voudrait tester son nouveau modèle d'extraction d'entités nommées sur un jeu de données en français avec des catégories liées à la géographie.
Ex. 2	Fabrice souhaite intégrer dans son moteur de recherche sur les annonces les scores de confiance pour trier les résultats et faire remonter ceux pour lesquels la confiance est maximale.
QC	- Quelles sont les annonces avec des entités appartenants à la catégorie « Toponym » et ayant une confiance supérieure à 0.8 ? - Quelles sont les annonces dont les entités extraites ont toutes une confiance supérieur à 0.5 ?

TABLE 4 – Scénario 4 : Utilisation d'annotations textuelles

2 État de l'art

Dans cette section, nous discutons les ontologies et graphes de connaissances existants pour le domaine de l'immobilier et les données spatiales, ainsi que leurs limites à la lumière de nos scénarios. La construction d'une ontologie pour l'immobilier dépend du point de vue adopté et du type de données utilisées. Dans [10], les auteurs comparent plusieurs ontologies appliquées à l'immobilier selon plusieurs perspectives : le territoire et notamment le cadastre [11], les transactions [12] et la juridiction [13]. L'ontologie *proDataMarket* [14] regroupe les trois domaines précédents et permet d'étudier le marché immobilier au travers des parcelles et des transactions. Néanmoins, cette ontologie se base sur des données anciennes (par exemple *Demande de Valeur Foncière* [1] en France) et étudie seulement les parcelles. Ainsi, il n'est pas possible de rechercher des biens immobiliers récents et en vente selon leurs caractéristiques intérieures (nombre de pièces, étage, etc.) ou leur environnement. L'ontologie *NAREO* [15] s'intéresse plutôt à l'environnement d'un bien immobilier avec la description des services et aménités à proximité du quartier dans lequel le bien est localisé. L'application de cette ontologie est la recommandation de quartiers selon des critères de localisation et d'environnement. Cependant, les auteurs ne décrivent pas les caractéristiques du bien et utilisent des jeux de données officiels tels que *OpenStreetMap* et les données de l'*INSEE*. Ni la perception de l'agent immobilier dans la description d'un lieu de vie, ni les noms de lieux vernaculaires ne sont pris en compte. La dimension spatiale des données est primordiale en immobilier puisqu'un bien immobilier est localisé sur le territoire (par exemple sur une parcelle) et que sa localisation joue un rôle prépondérant dans la décision d'achat. De nombreux graphes de connaissances ont intégré des entités spatiales tels que *DBPedia* [19], *Yago2Geo* [18], *WorldKG* [17] ou *KnowWhereGraph* [16]. Les données de ces graphes proviennent principalement d'agences gouvernementales (*INSEE*, *IGN*) ou de projets participatifs comme *Wikipedia* ou *OpenStreetMap*. Cependant, dans notre étude nous avons des données

1. https ://app.dvf.etalab.gouv.fr/

qui ne sont pas toujours répertoriées dans ces sources, ce qui limite leur utilisation. Différentes approches ([20]) ont été développées pour extraire des lieux vernaculaires (i.e., locaux, non-officiels) sur le Web et ainsi enrichir les gazetteers, mais ne proposent pas une manière standard de représenter le concept de lieu. D'un point de vue ontologique, il existe plusieurs manières de décrire une entité spatiale. *GeoNames*[2] utilise les concepts *SKOS* pour décrire des classes haut-niveau. *GeoLinkedData*[3] définit trois ontologies selon le domaine d'utilisation (administratif, transport, hydrographie) en réutilisant des vocabulaires existants. Enfin, l'*IGN*[4] a développé sa propre ontologie en s'appuyant sur la *BDTOPO* pour décrire les entités topographiques et administratives du territoire (bâtiment, réseau routier, végétation, etc.). Les limites de ces ontologies sont qu'elles classent toutes les entités selon leur nature et leur topographie. Or, notre application suggère une représentation des entités spatiales selon leur perception et leur utilisation. Dans [21] et [22], les auteurs proposent une représentation du concept de lieu notamment en mettant l'accent sur la formalisation de la provenance des informations et leur date d'utilisation. Cependant, ces représentations restent limités pour les lieux cognitifs. Finalement, pour rassembler les communautés du Web Sémantique et de la Géographie, *GeoSPARQL*[5] a été développé pour représenter et requêter les données spatiales. Son ontologie, composée de trois classes haut-niveau (*SpatialObject*, *Feature* et *Geometry*), offre une grande flexibilité pour décrire des entités spatiales et leurs géométries selon le domaine d'application. Celle-ci se base sur un ensemble de standards du Simple Feature Access[6] qui définit (1) une architecture commune pour la géométrie et sa représentation en text (WKT) ainsi que (2) un extension spatial des fonctions SQL.
En résumé, le but de ce travail est de représenter à la fois les informations immobilières des annonces et les données spatiales incertaines en un seul graphe de connaissance.

3 Localisation imprécise des lieux

Les scénarios et questions précédentes expriment le besoin de représenter les limites des lieux extraits dans les annonces immobilières, et plus particulièrement les limites des lieux vernaculaires (i.e., propres à la région étudiée) et non-officiels. En effet, la localisation et l'environnement du bien sont en général décrits dans les annonces immobilières mais au travers du regard de l'agent immobilier. Ainsi, les limites des lieux mentionnés sont celles perçues par l'agent immobilier et peuvent être différentes des limites administratives ou bien exagérées dans le but de vendre le bien [6]. Ces limites ne peuvent donc pas être simplement représentées par un point ou un polygone et demandent une représentation intégrant cette imprécision. Pour cela, nous avons décidé d'utiliser la théorie des ensembles flous.
Dans la théorie des ensembles flous [5], un sous-ensemble flou A d'un ensemble E est caractérisé par une application appelée fonction d'appartenance et notée μ_A. Celle-ci donne le degré d'appartenance à l'ensemble flou A, pour chaque élément x de E. Le degré d'appartenance est généralement dans l'intervalle [0,1]. On dit que si $\mu_A(x) = 1$, alors x appartient totalement à A tandis que si $\mu_A(x) = 0$, alors x n'appartient pas du tout à A.
Nous appliquons cette théorie aux lieux extraits pour capturer une approximation de la localisation en calculant le degré d'appartenance de chaque point de l'espace. De plus, il est possible d'obtenir des limites nettes (par exemple pour projeter la localisation sur une carte) en utilisant les α-coupes. Une α-coupe d'un ensemble flou A notée $\tilde{A}_\alpha$ est un sous-ensemble net dont chaque élément a un degré d'appartenance supérieur ou égal à α :

$$\tilde{A}_\alpha = \{x \in A; \mu_{\tilde{A}}(x) \geq \alpha\}.$$

Le noyau et le support sont des α-coupes particulières pour lesquelles α est égal respectivement à 1 et 0 :

$$\begin{aligned} noy(A) &= \{x \in A; \mu_A(x) = 1\}, \\ supp(A) &= \{x \in A; \mu_A(x) > 0\}. \end{aligned}$$

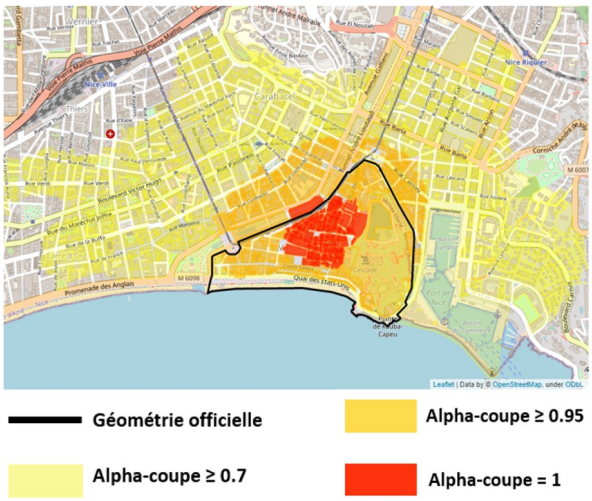


FIGURE 1 – Exemple de notre représentation incertaine VS les limites officielles du Vieux Nice

Dans l'étude des annonces immobilières, nous avons extrait un grand nombre de lieux vernaculaires qui sont mentionnés dans des annonces géolocalisées précisément (i.e., la latitude et la longitude sont sur ou proche du bâtiment). Nous avons donc pu estimer les limites d'un lieu en appliquant d'abord une méthode d'estimation par noyau gaussien (KDE) sur les coordonnées des annonces précisément géolocalisées et évoquant ce lieu. Cette méthode retourne une densité de probabilité sur l'espace étudié qui peut être facilement transformée en fonction d'appartenance d'un ensemble flou. La figure 1 montre un exemple de représentation incertaine du quartier du Vieux Nice calculée à

2. http ://geonames.org/ontology/ontology_v3.0.rdf
3. http ://geo.linkeddata.es
4. http ://data.ign.fr/def/topo/20190212.htm
5. http ://www.opengeospatial.org/standards/geosparql
6. https://www.ogc.org/standard/sfa/

partir des annonces immobilières et sa géométrie officielle. Pour cet exemple, nous avons représenté la géométrie incertaine à partir de 3 alpha-coupes projetées sur les parcelles de Nice. En effet, les biens immobiliers se trouvent sur des parcelles donc nous donnons un degrés d'appartenance à un lieu pour chaque parcelle. Nous pouvons remarquer que notre noyau ($\alpha = 1$) se situe dans la partie du quartier où les habitations sont très anciennes tandis que le reste du quartier a un degré d'appartenance un peu plus faible. Enfin, nous pouvons remarquer que le quartier du Vieux Nice a une grande influence puisqu'il est cité au-delà de ses limites officielles.

4 Modélisation ontologique

Dans cette section, nous décrivons et discutons la modélisation choisie pour définir les entités et les relations de notre ontologie. La figure 2 montre un exemple de notre modélisation appliquée à une annonce immobilière. Nous utilisons le préfixe `sure:` dans le reste de la section pour faire référence à l'ontologie *SURE*[7] (Spatial Uncertainty and Real Estate) que nous avons développée et publiée selon les standards et bonnes pratiques des données liées. Cette ontologie contient des classes et propriétés relatives au domaine d'étude et décrites ci-dessous, ainsi que des classes générées à partir des textes des annonces immobilières.

4.1 Représentation du bien immobilier

Les deux premiers scénarios (1,2) nous permettent d'identifier les entités et relations liées au bien immobilier. Nous avons notamment besoin de modéliser les termes suivants :

1. Le bien et son type ;
2. Les caractéristiques du bien (prix, étage, surface, nombre de pièces, calme, rénové, etc.) ;
3. La localisation du bien : la ville, les coordonnées et les lieux mentionnés dans l'annonce avec les relations spatiales.

Nous avons utilisé les vocabulaires *GeoSPARQL* et *schema.org* pour représenter le bien immobilier. La classe *Accommodation* de *schema.org* nous permet de modéliser le type du bien (*Apartment*, *House*, etc.) et certaines caractéristiques du bien grâce aux propriétés préalablement définies (*numberOfRooms*, *floorLevel*, etc.). D'autre part, le bien immobilier est aussi un objet spatial puisqu'il peut avoir des coordonnées ou des relations spatiales avec des lieux. Nous avons donc choisi de créer une sous-classe *RealEstate* de la classe *Feature* de *GeoSPARQL*. Ceci nous permet de créer une géométrie si nous connaissons sa position ou d'utiliser les propriétés de *GeoSPARQL* pour le localiser dans la ville et dans les lieux extraits (*geo:sfWithin*).

4.2 Représentation des lieux et localisations

Le troisième scénario (tableau 3) décrit les besoins relatifs à la représentation des lieux géographiques. Néanmoins, pour modéliser ces lieux, nous devons d'abord définir ce qu'est un lieu. Dans [7], les auteurs décrivent quatre manières de parler d'un lieu :

- *Place-Names* : la manière la plus simple de parler d'un lieu est d'utiliser son nom propre (Nice, Promenade des Anglais) ;
- *Place-Like Count Nouns* : utilisation d'un nom commun pour lequel un objet peut être localisé "dans" (ville, quartier, environnement, etc.) ;
- *Locative Property Phrases* : composition d'un nom propre ou commun avec une relation spatiale ("proche de la promenade des Anglais", "non loin de la gare", etc.) ;
- *Definite Descriptions* : description d'un objet à partir d'un autre objet et d'une relation spatiale ("la rue derrière la gare", "l'école à côté de la place Masséna").

D'autres travaux ([8], [9]) définissent seulement deux catégories pour parler d'un lieu : lieu absolu et lieu relatif. Le lieu absolu s'apparente à la catégorie *Place-Names* tandis que le lieu relatif est composé d'un lieu absolu et d'une relation spatiale.

Dans notre approche, nous avons décidé de créer un classe générale *Place* et une sous-classe *RelativePlace* qui correspond aux lieux composés d'une relation spatiale autre que "dans". La classe *Place* est une sous-classe de *geo:Feature*. La classe *RelativePlace* a deux propriétés supplémentaires permettant de définir le type de relation spatiale et l'objet de la relation spatiale (*hasSpatialRelation*, *hasAnchor*).

Les instances de *Place* sont principalement les lieux définis avec des noms propres (Place Masséna, Nice, etc.). Néanmoins, les noms communs qui s'apparentent à des lieux et dans lesquels le bien peut être localisé (centre-ville, zone piétonne, rue, etc.) sont aussi des instances de cette classe. Pour choisir si un nom commun peut être vu comme un lieu ou non, nous avons créé deux classes, *Amenity* et *LocativeArea*, qui sont aussi des sous-classes de *geo:Feature*. *LocativeArea* regroupe les noms communs qui localisent le bien. *Amenity* regroupe les entités qui se trouvent à un certaine proximité du bien et qui lui donnent de la valeur (gare, école, port, plage, etc.). Enfin, les lieux composés d'un nom propre ou d'un nom commun (*LocativeArea* ou *Amenity*) et d'une relation spatiale sont des instances de la classe *RelativePlace*. Nous avons généré automatiquement des sous-classes de *LocativeArea* et *Amenity* à partir du texte des annonces immobilières (*Quartier*, *Gare*, *Rue*, etc.). A ce stade, aucun traitement n'a été appliqué a posteriori, à l'exception d'une heuristique requérant au moins deux instances pour qu'une classe soit conservée i.e. deux annonces mentionnant la classe. Une perspective sera de travailler sur l'amélioration de la qualité de cette extraction.

Enfin, nous représentons les géométries de ces lieux à l'aide de la théorie des ensembles flous présentée dans la section 3. Nous avons choisi de créer une classe *AlphaCut* qui est une sous-classe de *geo:Geometry*. Cette classe a la propriété *hasAlpha* pour définir le degré d'appartenance correspondant. Enfin, *GeoSPARQL* permet d'associer une collection de géométries à un même objet ce qui nous permet d'associer plusieurs *AlphaCut* à un même lieu pour re-

7. `http://ns.inria.fr/sure#`

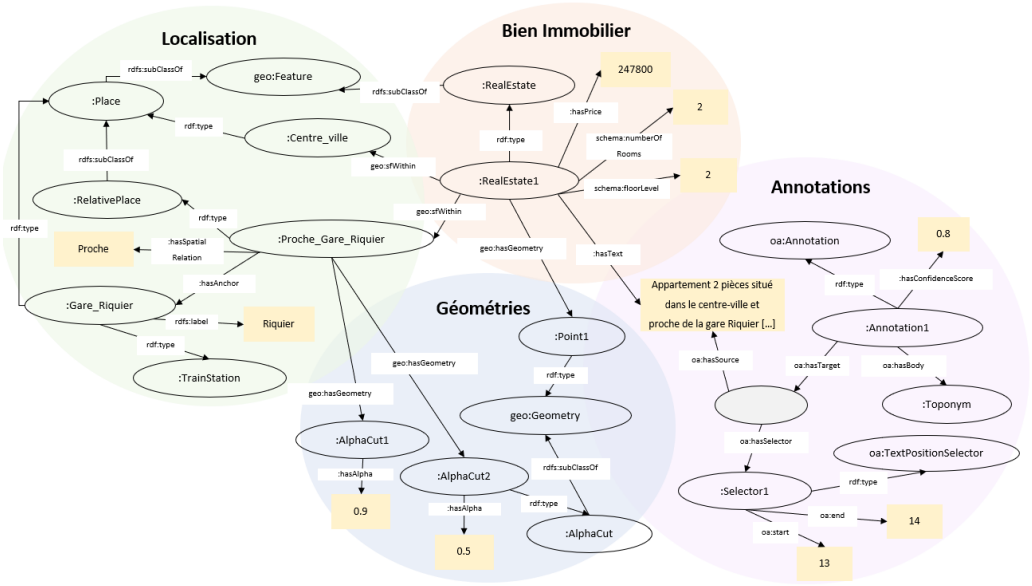


FIGURE 2 – Exemple d'un graphe RDF représentant les informations issues d'une annonce immobilière

présenter de manière aussi fiable que possible sa frontière floue. Le listing 1 donne un extrait de cette formalisation.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns
    #> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix : <http://ns.inria.fr/sure#> .

############# Classes et Proprietes
:TrainStation rdfs:subClassOf geo:Feature.
:Place rdfs:subClassOf geo:Feature.
:RelativePlace rdfs:subClassOf :Place.
:AlphaCut rdfs:subClassOf geo:Geometry.

:hasAlpha a rdfs:Property ;
   rdfs:domain :AlphaCut ;
   rdfs:range xsd:double .

:hasAnchor a rdfs:Property ;
   rdfs:domain :RelativePlace ;
   rdfs:range geo:Feature.

:hasSpatialRelation a rdfs:Property ;
   rdfs:domain :RelativePlace ;
   rdfs:range xsd:string.

############# Instances
:Gare_Riquier a :TrainStation, :Place;
  rdfs:label "riquier"@fr.

:Proche_Gare_Riquier a :RelativePlace;
  :hasAnchor :Gare_Riquier;
  :hasSpatialRelation "proche";
  geo:hasGeometry :AlphaCut1, :AlphaCut2.

:AlphaCut1 a :AlphaCut ;:hasAlpha "0.5"^^xsd:double;
    geo:asWKT "MULTIPOLYGON (((43.6957 7.280889,..., 43.
     69578 7.280882)))"^^geo:wktLiteral.

:AlphaCut2 a :AlphaCut ;:hasAlpha "0.9"^^xsd:double;
    geo:asWKT "MULTIPOLYGON (((43.695 7.2808,..., 43.
     6955 7.280892)))"^^geo:wktLiteral.
```

LISTING 1: Exemple de la syntaxe RDF d'un lieu incertain et de ses limites floues.

4.3 Représentation des textes annotés

Le dernier scénario (tableau 4) a un objectif qui ne touche pas à l'immobilier. Nous proposons d'utiliser les annotations textuelles produites par le modèle de reconnaissance d'entités dans le texte comme un jeu de données réutilisable pour mener d'autres recherches en traitement du langage naturel, notamment sur l'extraction d'entités nommées. Le vocabulaire *Web Annotation Data Model* permet d'annoter les entités retrouvées dans un texte. Nous avons donc fait le choix d'utiliser ce vocabulaire. Néanmoins, les annotations ne sont pas certaines puisqu'elles proviennent des prédictions du modèle de reconnaissance d'entités. Ainsi, nous avons ajouté le score de confiance donné par le modèle à l'annotation à l'aide d'une nouvelle propriété *hasConfidenceScore*.

5 Conclusion et Perspectives

La représentation des données immobilières issues de l'extraction d'information des annonces présente plusieurs enjeux que nous avons décrits dans cet article. Nous avons proposé une modélisation ontologique et justifié nos choix à l'aide de nos scénarios motivants. Ainsi, nous avons modélisé le bien immobilier, ses caractéristiques et sa localisation. Nous avons montré que la localisation est en général incertaine et nécessite une représentation particulière. Nous avons proposé d'utiliser la théorie des ensembles flous et d'intégrer les alpha-coupes dans notre ontologie. Enfin, nous avons ajouté les annotations textuelles issues d'un modèle de reconnaissance d'entités nommées afin de créer un jeu de données réutilisable pour mener d'autres recherches en traitement du langage naturel. La prochaine étape de ce travail est donc le peuplement de l'ontologie et la création du graphe de connaissances à partir des annonces immobilières localisées dans les Alpes-Maritimes

dans un premier temps, et dans toute la France dans un second temps. Nous nous attacherons aussi à évaluer notre modélisation. Finalement, le graphe de connaissances produit pourra permettre de retrouver des biens immobiliers similaires et créer, à termes, un système de recommandation.

Références

[1] Uschold, Mike, and Michael Gruninger. "Ontologies : Principles, methods and applications." The knowledge engineering review 11.2 (1996) : 93-136.

[2] Bosvieux, Jean. "L'immobilier, poids lourd de l'économie", Constructif, vol. 49, no. 1, 2018, pp. 10-14.

[3] Bekoulis, Giannis, Deleu, Johannes, Demeester, Thomas, and Develder, Chris. 2017. "Reconstructing the house from the ad : Structured prediction on real estate classifieds." In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers, pages 274–279, Valencia, Spain.

[4] Lucie Cadorel, Alicia Blanchi, and Andrea G. B. Tettamanzi. 2021. "Geospatial Knowledge in Housing Advertisements : Capturing and Extracting Spatial Information from Text." In Proceedings of the 11th on Knowledge Capture Conference (K-CAP '21). ACM, USA, 41–48.

[5] L.A. Zadeh. 1965. "Fuzzy sets." Information and Control 8, 3 (1965), 338–353.

[6] Grant McKenzie and Yingjie Hu. 2017. "The "Nearby" Exaggeration in Real Es- tate (Position Paper)." In Proceedings of the Cognitive Scales of Spatial Information Workshop (CoSSI 2017) (L'Aquila, Italy), Werner Kuhn, Dan Montello, Scott Fre- undschuh, Crystal Bae, Thomas Harvey, Sara Lafia, and Daniel Phillips (Eds.). 4–8.

[7] Bennett, B., Agarwal, P. (2007). Semantic Categories Underlying the Meaning of 'Place'. In : Winter, S., Duckham, M., Kulik, L., Kuipers, B. (eds) Spatial Information Theory. COSIT 2007. LNCS, vol 4736. Springer.

[8] Lesbegueries, Julien, Christian Sallaberry and Mauro Gaio. "Associating spatial patterns to text-units for summarizing geographic information." Workshop on Geographic Information Retrieval (2006).

[9] Syed, M. A., Arsevska, E., Roche, M., and Teisseire, M. : GeoXTag : Relative Spatial Information Extraction and Tagging of Unstructured Text, AGILE GIScience Ser., 3, 16,

[10] Ling Shi and Dumitru Roman. 2018. Ontologies for the Real Property Domain. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics (WIMS '18). Association for Computing Machinery, New York, NY, USA, Article 14, 1–8.

[11] D Sladić, M Govedarica, D Pržulj, A Radulović and D Jovanović (2013) Ontology for real estate cadastre, Survey Review, 45 :332, 357-371, DOI : 10.1179/1752270613Y.0000000042

[12] Erik Stubkjaer. 2017. The ontology and modelling of real estate transactions. Routledge

[13] Jesper M Paasch. 2005. Legal Cadastral Domain Model : An Object-orientated Approach. Nordic journal of surveying and real estate research 2, 1 (2005), 117–136.

[14] Ling Shi, Nikolay Nikolov, Dina Sukhobok, Tatiana Tarasova, and Dumitru Roman. 2017. The proDataMarket Ontology for Publishing and Integrating Crossdomain Real Property Data. journal Territorio Italia. Land Administration, Cadastre and Real Estate 2 (2017)

[15] Wissame Laddada, Fabien Duchateau, Franck Favetta, and Ludovic Moncla. 2020. Ontology-Based Approach for Neighborhood and Real Estate Recommendations. In Proceedings of the 4th ACM SIGSPATIAL Workshop on Location-Based Recommendations, Geosocial Networks, and Geoadvertising (LocalRec'20). ACM, USA, Article 4, 1–10.

[16] Krzysztof Janowicz, Pascal Hitzler, Wenwen Li, Dean Rehberger, Mark Schild- hauer, Rui Zhu, Cogan Shimizu, Colby K Fisher, Ling Cai, Gengchen Mai, et al. 2022. Know, Know Where, KnowWhereGraph : A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environ- mental intelligence. AI Magazine 43, 1 (2022), 30–39.

[17] Dsouza, Alishiba and Tempelmeier, Nicolas and Yu, Ran and Gottschalk, Simon and Demidova, Elena. WorldKG : A World-Scale Geographic Knowledge Graph. 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021.

[18] Nikolaos Karalis, Georgios M. Mandilaras, and Manolis Koubarakis. 2019. Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge. In Proc. of the ISWC 2019, Part II (Lecture Notes in Computer Science, Vol. 11779). Springer, 181–197

[19] Sö ren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia : A Nucleus for a Web of Open Data. In Proc. of the ISWC 2007 (Lecture Notes in Computer Science, Vol. 4825). Springer, 722–735

[20] Christopher B. Jones, Ross S. Purves, Paul Clough, Hideo Joho : Modelling Vague Places with Knowledge from the Web. International Journal of Geographic Information Science 22(10), 1045 – 1065 (2008)

[21] Karl Grossner and Ruth Mostern. "Linked places in world historical gazetteer." Proceedings of the 5th ACM SIGSPATIAL International Workshop on Geospatial Humanities. 2021

[22] Andrea Ballatore. "Prolegomena for an ontology of place". Advancing geographic information science, 91-103. 2016

Session 2 : Modélisation des connaissances

Un patron de conception pour la modélisation ontologique des paradigmes expérimentaux

J. Hilbey[1,2], X. Aimé[3,2], J. Charlet[4,2]

[1] Sorbonne Université, Paris, France
[2] Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en e-Santé, LIMICS, Paris, France
[3] Cogsonomy, Nantes, France
[4] Assistance Publique-Hôpitaux de Paris, Paris, France

jacques.hilbey@sorbonne-universite.fr

Résumé

Nous présentons un patron de conception ontologique pour la modélisation d'expériences scientifiques et d'examens menés lors d'une étude de recherche clinique. Intégrer des données hétérogènes dans un modèle ontologique commun est un défi, redoublé par l'exigence de pouvoir les explorer ultérieurement. Afin de faciliter l'élaboration des modules ontologiques dédiés, ce patron de conception s'appuie sur des invariants, est centré sur l'événement de la passation, et conserve le lien aux données originales.

Mots-clés

Patron de conception ontologique, ontologies biomédicales, recherche biomédicale.

Abstract

We present an ontology design pattern for modeling scientific experiments and examinations conducted in a clinical research study. Integrating heterogeneous data into a common ontological model is a challenge, redoubled by the requirement to be able to explore them later. In order to facilitate the development of dedicated ontological modules, this design pattern relies on invariants, is centered on the event of the test, and keeps the link to the original data.

Keywords

Ontology Design Pattern, Biomedical Ontologies, Biomedical Research.

1 Introduction

Le projet PsyCARE [1] est un projet de Recherche Hospitalo-Universitaire en santé qui vise à améliorer la détection et l'intervention précoce en cas de psychose, et à offrir des programmes thérapeutiques personnalisés. Afin d'atteindre cet objectif, le projet se propose :

- d'identifier des biomarqueurs pour améliorer le diagnostic, la détection du stade de la maladie et la prédiction du devenir fonctionnel ;
- d'identifier une liste de cibles pour les stratégies de modification de la maladie ;
- de fournir et de valider :
  - une application pour l'entraînement cognitif personnalisé,
  - une application centrée sur le patient facilitant la gestion des cas et l'engagement du patient,
  - un système d'aide à la décision qui guidera la stratégie thérapeutique personnalisée ;
- de développer une plateforme innovante de collecte et d'intégration des données du projet adaptée à la recherche en psychiatrie et aux soins en santé mentale ;
- de fournir un kit d'outils pour la diffusion des connaissances et la formation afin d'améliorer la sensibilisation et le transfert des résultats de PsyCARE à la pratique médicale.

Différentes études sont mises en œuvre dans le cadre du projet, dont la principale est multicentrique. La plateforme de collecte des données est le point d'accès unique, pour les utilisateurs finaux, aux données transmises par les producteurs de données.

Le modèle conceptuel de la plateforme de collecte des données est fourni par une ontologie modulaire dont les modules de domaines correspondent aux données hétérogènes recueillies : données cliniques, d'imagerie cérébrale, de biologie et de biologie moléculaire, d'analyse de la parole, d'évaluation motrice.

Le rôle que peuvent jouer les ontologies dans l'intégration et l'échange de données en permettant une interopérabilité sémantique est bien établi [2].

Afin de faciliter l'intégration de ces données dans les différents modules ontologiques dédiés, la conservation de métadonnées de provenance, l'exploration des données et leur réutilisabilité ultérieure, nous proposons dans cet article un patron de conception ontologique centré sur les événements de passation d'expérience applicable largement aux paradigmes expérimentaux de différentes disciplines.

1. https://psy-care.fr/

2 Matériel et méthodes

2.1 Les sources de données

Outils d'évaluations psychiatrique Les outils d'évaluation utilisés en psychiatrie permettent de déterminer, en l'absence de biomarqueur fiable de la pathologie, le seuil entre le normal et le pathologique. Ils prennent la forme de questionnaires ou d'autoquestionnaires, d'entretiens, de tests ou de tâches, parfois standardisés et validés (on parle alors d'échelles).

Examens d'imagerie cérébrale Des examens d'imagerie par résonance magnétique (IRM) anatomique ou fonctionnelle, d'électroencéphalographie (EEG), de tomodensitométrie par émission de postions (TEP) du cerveau, sont menés pour guider les interventions thérapeutiques.

Examens de biologie médicale Différents examens de biologie médicale sont effectués afin de permettre l'identification de biomarqueurs biologiques, là aussi pour guider les interventions thérapeutiques.

Examens de biologie moléculaire Des examens de biologie moléculaire sont mis en œuvre afin d'identifier des biomarqueurs génétiques des symptômes ou des troubles d'intérêt, des mutations génétiques constituant des prédispositions, des mécanismes étiologiques sous-jacents suggérant des interventions thérapeutiques.

Analyse de la parole Des enregistrements de sessions thérapeute-patient sont analysés d'une part dans leurs aspects prosodiques, puis après transcription dans leurs aspects syntaxiques et lexicaux par des techniques de Traitement Automatique des Langues.

Evaluation des troubles de la dextérité et des fonctions cognitivo-motrices Différentes tâches effectués sur une tablette numérique sont proposées au sujet de l'expérimentation, mettant en jeu la précision et la rapidité des mouvements des doigts, la rotation mentale, la capacité d'effectuer des mouvements isolés des doigts en évitant des syncinésies, la temporalité et la variabilité des mouvements fins des doigts, tâches effectuées sur une tablette numérique.

2.2 Éléments de description communs

Les données que nous venons de présenter sont hétérogènes à la fois par leur contexte de production, par les modèles scientifiques qui les sous-tendent, par les usages de formatage et de transmission qui s'y appliquent, et même par le lexique développé par chaque domaine pour désigner ses objets et ses méthodes. Nous pouvons toutefois pointer les éléments communs à ces processus de production de données : **l'objet d'étude**, **la passation d'une expérience**, expérience qui obéit pour les différents objets d'étude à un même **protocole** et qui génère en premier lieu un **produit concret** à partir duquel sont effectuées des **mesures** ou des évaluations. Le protocole, antérieur à la réalisation des expériences, prévoit nécessairement un **instrument de captation** du produit concret, et implique éventuellement une **mise en condition** de l'objet d'étude ou de l'agent de passation permettant d'assurer la comparabilité des résultats obtenus, ainsi qu'un ou des **stimulus** permettant de conditionner le phénomène étudié pour en affiner l'observation.
Le tableau 1 montre comment les différents éléments communs retenus sont déclinés selon les différentes sources de données. Le terme d'« instruction » recouvre ce que nous avons appelé plus haut « mise en condition ».

Le repérage de ces éléments de description communs ne dispense pas d'une réflexion sur la manière dont ils apparaissent dans les domaines. Nous pensons notamment à la comparabilité de la granularité des examens, des séquences, des expériences, et des événements concrets qui y sont associés.

2.3 Engagement ontologique

Les modules ontologiques développés pour les différentes sources de données s'inscrivent dans une ontologie modulaire pour laquelle ont été développées antérieurement une ontologie fondationnelle (ontoPOF) et une ontologie noyau des données médicales (ontoDOME). Certains engagements ontologiques pris lors de leur élaboration permettent d'expliciter certains aspects du patron de conception ontologique présenté dans cet article.

ontoPOF est une ontologie fondationnelle qui préserve une forte compatibilité avec les ontologies fondationnelles endurantistes comme BFO ou DOLCE, mais :

- ouvre des possibilités de représentation de la dynamique temporelle des entités définies par l'espace qu'elles occupent [6] (les « objets » en un sens large, incluant les êtres vivants) en les reliant systématiquement à leur existence entière, aux événements auxquels elles participent et aux événements dont elles sont le lieu ;
- spécifie la relation de participation à un événement pour indiquer le rôle tenu ;
- considère les projets, entités intentionnelles, comme des entités de premier plan et modélise leur réalisation dans un ou plusieurs événements.

D'une manière générale, cette ontologie fondationnelle est centrée sur les événements considérés comme les seules entités primitives concrètes, afin de permettre la représentation de la dynamique temporelle (l'évolution) des entités présentes dans l'ontologie.

ontoDOME est une ontologie noyau pour le domaine de la santé qui modélise les connaissances telles qu'elles sont échangées ou décrites dans des documents médicaux, plutôt que telles qu'elles existeraient dans l'esprit des experts médicaux. Ce choix :

- repose sur une vision des ontologies comme des artefacts numériques qui étendent la cognition humaine, tant pour les producteurs de données – qui peuvent stocker de grandes quantités de données dont la cohérence est assurée par le modèle –, que pour les utilisateurs finaux – qui peuvent explorer ces mêmes données sans connaître le modèle de stockage des données sous-jacent ;
- favorise la traçabilité des données et, plus généralement, le respect des principes FAIR (les docu-

Domaine	**Psychométrie**	**Imagerie**	**Biologie**	**Biologie moléculaire**	**Analyse du discours**	**Dextérité**
Expérience (passation)	Questionnaire, entretien, test	Séquence IRM, EEG, TEP	Prélèvement en laboratoire	Prélèvement en laboratoire	Enregistrement	Tâche de motricité
Stimulus	Question, dessin, etc.	Son, image, etc.	NA	NA	Question	Images, sons
Outil de captation	Papier et crayon	Appareil IRM, EEG, TEP	Kit de prélèvement	Kit de prélèvement	Appareil enregistreur	Tablette numérique
Instruction	au sujet, à l'agent de passation	Paramétrage, agent de contraste	au sujet, de prélèvement	NA	NA	au sujet
Produit	Réponse	Jeu d'images	Échantillon	Échantillon	Fichier audio	Gestes
direct Mesure	Scores	Volumes, régions d'intérêt	Valeurs, comparaisons à des seuils	Id. de gènes, de variants, de *pathways*	Valeurs de variables	Valeurs de variables

TABLE 1 – Éléments communs selon les différentes sources de données de PsyCARE

ments sont référencés et accessibles via un référentiel commun qui est validé sémantiquement et partagé par les experts médicaux).

Ces engagements ontologiques nous amènent, dans l'élaboration du patron de conception ontologique, à porter une attention particulière à :

— la conservation d'un parallélisme entre le projet (le protocole de l'expérience) et l'événement qui le réalise (la passation de l'expérience), afin de pouvoir comparer les deux ;
— l'introduction des données entre la passation de l'expérience et les mesures auxquelles elle donne lieu : ce sont les informations effectivement transmises entre les différents acteurs ; elles présentent différents niveaux de granularité, des données d'une étude aux données relatives à un item d'une collecte de données (reflétées dans la granularité des événements : étude, visite, examen, collecte de données, étape de collecte de données).

2.4 Les patrons de conception ontologique

Un patron de conception ontologique est une modélisation qui peut servir pour résoudre un problème de conception ontologique qui se pose de façon récurrente [7]. Dégager un tel patron repose donc sur l'observation d'invariants dans les données, les objets, les processus, les relations.

Les patrons de conception ontologique (ou *Ontology Design Patterns* – OP) peuvent être de plusieurs types [4] :

— les *Structural OPs*, qui subsument (i) les *Logical OPs* résolvant un problème d'expressivité et (ii) les *Architectural OPs* contraignant la constitution générale de l'ontologie ;
— les *Reasoning OPs* qui orientent le raisonnement du moteur d'inférence ;
— les *Correspondence OPs*, qui subsument (i) les *Reengineering OPs* transformant un modèle source en modèle ontologique et (ii) les *Mapping Ops* pour exprimer des correspondances ;
— les *Presentation OPs* pour les conventions de nommage et les schémas d'annotation ;
— les *Lexico-syntactic OPs* qui associent des formes linguistiques à une signification ;
— les *Content OPs* qui proposent des patrons pour le contenu d'un domaine particulier.

Cette dernière catégorie correspond très précisément à ce que nous proposons ici.

2.5 État de l'art

Une recherche par mots-clés sur le site web de l'*Association for Ontology Design & Patterns* (ODPA) [2] ne donne pas de réultat pour le sujet que nous traitons. En revanche, quelques ontologies proposent des modélisations d'expériences scientifiques ou d'examens cliniques.

EXPO est une ontologie s'appuyant sur une description formelle des expériences scientifiques afin de les annoter [9]. Elle utilise SUMO comme ontologie fondationnelle et entend couvrir tout le champ de la science expérimentale, en termes de types d'hypothèse, de modèle, d'expérience, de variable, de résultats. Elle se présente comme un ontologie de niveau intermédiaire entre SUMO et des ontologies de domaine consacrées à un champ d'étude particulier.
Sa finalité est donc en premier lieu la caractérisation d'expériences scientifiques, ce qui permet de les situer méthodologiquement dans le champ de la science. En ce sens, cette ontologie ne répond pas à notre cas d'usage.

CogPo est une ontologie qui entend décrire les paradigmes de psychologie cognitive [10]. On y trouve la plupart des éléments avancés dans la section 2.2 : instructions, stimulus, paradigme, condition (qui correspond à ce que nous ap-

2. http://ontologydesignpatterns.org/

pelons ici « protocole d'expérience »). Comme pour EXPO, l'enjeu est plus une caractérisation d'un paradigme dans un champ scientifique qu'une modélisation d'une expérimentation en action, qui génère des données. Elle est utilisée par l'*Ontology of Experimental Variables and Values* (**OOEVV**) pour modéliser les variables d'IRM fonctionnelle [3]. L'ontologie légère OOEVV, qui entend modéliser les variables et leurs valeurs et pouvoir se raccorder aux ontologies de l'OBO Foundry, n'a pas connu de développement dans les autres domaines qui nous intéressent, ni après 2012.

Ontology for Biomedical Investigations (OBI) est une ontologie des études cliniques et biomédicales [1] très détaillée (plus de 4 000 classes). Elle est issue de la *Functional Genomics Investigation Ontology* (FuGO, qu'EXPO présentait comme une ontologie de domaine qu'elle se proposait de subsumer), se place sous l'ontologie fondationnelle *Basic Formal Ontology* (BFO) et reprend des classes de nombreuses ontologies ressortissant de l'OBO Foundry, notamment l'*Ontology for General Medical Science* (OGMS) pour ce qui concerne les aspects médicaux (phénotype, maladie, diagnostic, traitement) et de l'*Information Artifact Ontology* (IAO) pour ce qui concerne les objets informationnels. L'objectif d'OBI étant de fournir un modèle ontologique qu'une étude biomédicale viendra instancier, elle répond déjà à de nombreux cas d'usage et est appelée à croître pour en embrasser plus.

En proposant un patron de conception ontologique facilitant l'élaboration de modules selon les domaines, nous opérons un choix différent : nous proposons une structure minimale permettant d'intégrer les données et de les explorer, et laissons aux modules de domaines une modélisation plus poussée du domaine. D'autre part, les niveaux de granularité des événements de l'étude, des données qu'ils génèrent et des éventuels protocoles correspondants, ne sont que marginalement pris en considération ; cette critique concerne toutefois plutôt IAO, voire BFO, qu'OBI en propre.

Hors du domaine biomédical, l'ontologie *Semantic Sensor Network* [5] (**SSN**) propose une modélisation très aboutie recoupant nos préoccupations, même si par construction elle est centrée sur les dispositifs d'acquisition (ou d'actionnement) quand nous nous centrons sur la connaissance acquise sur le sujet étudié.

3 Résultats

3.1 Un patron de conception ontologique pour les paradigmes expérimentaux

Un paradigme expérimental peut être considéré comme étant dans une relation d'abstraction par généralisation à différentes conditions expérimentales, qui en seraient les spécifications. Cette conception amènerait à une modélisation des conditions comme autant de sous-classes d'un paradigme expérimental. En prenant en compte la pratique expérimentale effective, qui mêle au sein d'une même session (unité temporelle) la passation de différentes conditions expérimentales obéissant au même paradigme, nous

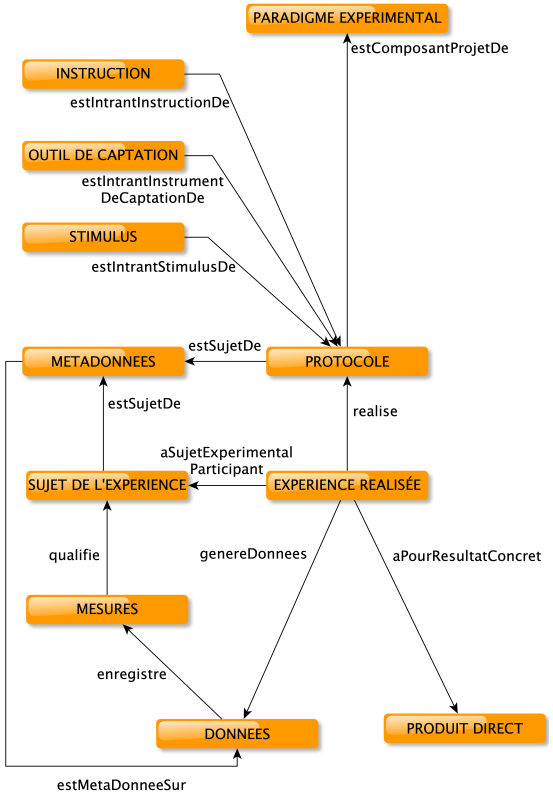


FIGURE 1 – **Un patron de conception d'ontologie pour les paradigmes expérimentaux.**

choisissons de considérer une relation méréotopologique de composition entre le paradigme et les conditions, reflétant la relation méréotopologique entre les événements (temps de la session, qui a pour parties les temps des différentes conditions).

La figure 1 présente les résultats au niveau de la condition. C'est à ce niveau que sont précisés, pour le protocole de la condition, quels sont les stimulus, instruction et outil de captation prévus en tant qu'intrants – le protocole est un projet qui est réalisé lors de chaque passation de l'expérience par un sujet.

Le patron de conception tient essentiellement dans les relations (*Object Properties*) qui sont indiquées. Les éléments de description communs que nous avons dégagés dans la section 2.2 ne doivent pas être conçus comme des classes nécessairement présentes dans l'ontologie.

Si certains éléments (nous nous plaçons ici au niveau des instances) présentent une proximité ontologique forte, comme les sujets, les événements de passation, les protocoles, ce qui peut amener à prévoir des classes (éventuellement des classes définies) pour les représenter dans la partie taxonomique de l'ontologie, d'autres peuvent recouvrir une forte diversité ontologique, comme les stimulus, les produits directs, les outils de captation ou les instructions. Qui plus est, ces derniers éléments peuvent selon les contextes apparaître sous différents aspects (une portion de sang prélevé peut faire l'objet d'analyses ou d'un don de sang ; un

son servant de stimulus peut entrer dans une composition musicale) et leur position dans la taxonomie dépendra de leur nature intrinsèque, mais c'est une relation à d'autres entités qui permettra d'indiquer leur fonction ou leur rôle dans le contexte qui nous intéresse. D'autre part, la position des éléments de description communs dans la taxonomie peut tenir à l'engagement ontologique pris en amont, comme nous l'avons montré dans la section 2.3.

Aux éléments communs spécifiés plus haut s'ajoutent les métadonnées, qui portent sur le protocole et le sujet de l'expérience, ce qui permet d'établir un lien plus direct entre les données et les informations importantes de leurs conditions de production : le sujet et le protocole spécifique de la condition expérimentale.

Certaines relations peuvent sembler redondantes. Elles permettent de raccourcir le chemin dans le graphe de connaissance, selon le mode d'exploration souhaité. Si l'on se centre sur le sujet de l'expérience, on peut accéder rapidement à l'étude dans laquelle l'expérience a été réalisée et aux mesures qui le qualifient, ou inscrire l'événement de la passation dans une chronologie des événements connus de la vie du patient ; si l'on se centre sur les données, on peut accéder rapidement à celles-ci et à leurs métadonnées ; si l'on se centre enfin sur l'événement de la passation de l'expérience (qui dans le graphe présenté est le nœud qui a la plus grande centralité de proximité aux autres nœuds), on peut l'inscrire dans une chronologie des événements de l'étude.

3.2 Application à la dextérité

Nous présentons dans la figure 2 une application de notre patron de conception ontologique au module consacré à la dextérité et à la psychomotricité :

- le paradigme général est un paradigme de reconnaissance des doigts (subdivisé en conditions Inverse, Miroir et Transversal) ;
- l'événement considéré est la passation d'une expérience de reconnaissance des doigts (*finger recognition*) en condition Inverse ;
- il obéit à un protocole qui met en jeu une tablette numérique comme instrument de captation des réactions psychomotrices du sujet et une image de main inversée comme stimulus (le patient est invité à poser le doigt qui est indiqué sur la représentation inversée d'une main) ;
- différentes mesures sont effectuées : le temps de réaction, le taux de réussite, le taux d'échec, et le taux de coactivation – qui mesure si d'autres doigts bougent en même temps que le doigt ciblé ;
- l'instrument de captation est conçu pour traduire immédiatement en mesures le produit direct (le geste du patient), qui n'est donc pas conservé ;
- aucune instruction n'est mentionnée dans le protocole, qui ne comporte donc pas ce type d'intrant.

3.3 Utilisation et validation

Ce patron de conception a été utilisé sur les différents modules de domaines. Il a permis d'orienter la modélisation en

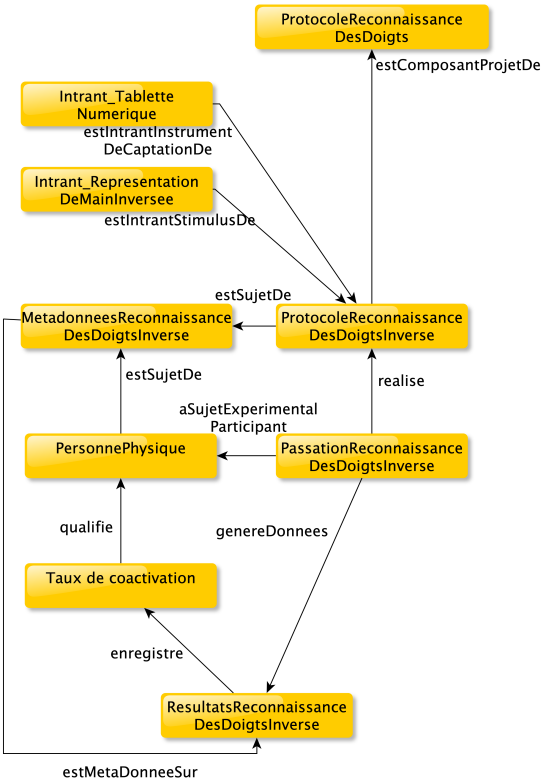


FIGURE 2 – **Application aux données de dextérité.**

poussant à identifier dans les différentes sources de données les éléments à intégrer dans les ontologies dédiées. L'élaboration ainsi facilitée des modules de domaine constitue une validation interne.

Il a été déposé sur le site de l'ODPA [3] afin de pouvoir être revu et critiqué par la communauté autour des *Ontology Design Patterns*.

Le module consacré à la dextérité, accompagné de requêtes SPARQL illustrant les possibilités d'exploration des données, est disponible [4].

4 Discussion

Selon les sources de données, le patron de conception que nous présentons doit être adapté. Par exemple :

1. dans le module des données relatives à la dextérité, certaines conditions sont passées dans une même séquence et les données qui en résultent représentent parfois une moyenne, parfois une soustraction entre résultats obtenus dans chaque condition ; dans ces cas, les données ne peuvent être reliées à la passation réalisant un protocole, mais doivent être reliées à la passation réalisant un paradigme (événement dont la passation réalisant un protocole est un épisode),

3. http://ontologydesignpatterns.org/wiki/Submissions:ExperimentalParadigmData
4. https://framagit.org/jacqueshilbey/ontodext-op

2. dans le module des données d'analyse du discours, le produit concret de la passation est un fichier audio, qui fait ensuite l'objet d'une transcription ; il serait envisageable de modéliser plus avant ce processus, mais le choix effectué a été de conserver l'enregistrement et sa transcription comme deux produits concrets de la passation et de fusionner les données produites.

D'autre part, la modélisation que nous avons présentée dans la figure 1 se situe au niveau de l'expérience, considérée comme la plus petite entité, parmi les événements, présentant une unité (elle n'est pas composée, même si elle peut avoir des parties). Aux niveaux de granularité plus élevés, on retrouve la relation *realise* entre l'événement de passation d'un examen (correspondant au niveau du paradigme) et le protocole plus générique du paradigme, ainsi que la relation *genereDonnees* entre ce même événement et les résultats considérés globalement de l'examen (ainsi que les relations méréotopologiques par type d'entités – événements, protocoles, données – entre ces différents niveaux). L'utilisation de ce patron de conception n'exonère pas d'une identification précise de ces différents niveaux de granularité selon les différentes sources de données. Cette identification, comme nous l'avons expliqué dans la section 2.3, se fonde sur les événements, en recherchant ce qui reste invariant (unité de sujet de l'expérience, unité de lieu et de temps, unité de domaine d'exploration, unité de paradigme, unité de condition).

La traçabilité des flux de données et des traitements qui leur ont été appliqués, non évoquée ici, est assurée par l'importation d'une partie de l'ontologie BMS-LM [8] développée par l'industriel assurant la mise en place de la plateforme (Fealinx), et s'appuie sur PROV-O.

En termes d'exploration du graphe de connaissance, ce qui est privilégié est :

— l'analyse exploratoire des données, à partir des variables et du sujet qu'elles qualifient ;
— l'examen des données relatives à un cas ;
— la reconstitution de la chronologie des événements d'une étude.

5 Conclusion

Le patron de conception que nous présentons s'appuie sur la structure commune fournie par la méthodologie scientifique et sur l'importance des données dans la science contemporaine pour surmonter le défi que représente l'intégration de données hétérogènes. Il ne constitue pas une solution clé en main au sens où il oblige à se poser des questions concernant la granularité des événements impliqués, la place donnée aux produits concrets, les besoins de métadonnées. Une fois ces questions résolues, il permet un cadre unifié pour intégrer et explorer les données tout en conservant un souci de traçabilité de celles-ci. Ce cadre est conçu comme compréhensif : tous les éléments présentés ne sont pas nécessairement instanciés (voir le tableau 1 et la section 3.2), selon les sources de données et selon les cas d'usage.

Il est assez indépendant de l'ontologie de haut niveau sous laquelle on se place, même s'il suppose qu'en amont des ontologies de domaine y recourant, les ontologies de plus haut niveau permettent de modéliser des objets, des projets, des événements, des propriétés, des objets informationnels. Autant que possible, les éléments sont alignés à des vocabulaires ou a des terminologies de référence (LOINC pour les examens de biologie, NDA pour les échelles en psychiatrie).

Remerciements

Ce travail a bénéficié d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche au titre du Programme d'Investissements d'Avenir portant la référence PsyCARE ANR-18-RHUS- 0014.

Références

[1] A. Bandrowski, R. Brinkman, M. Brochhausen, M.H. Brush, B. Bug, M.C. Chibucos, et al., The Ontology for Biomedical Investigations. *PLoS ONE*, Vol. 11, 2016.

[2] O. Bodenreider, Biomedical Ontologies in Action : Role in Knowledge Management, Data Integration and Decision Support, *Yearb Med Inform.*, pp. 67-79, 2008.

[3] G.A.P.C. Burns, J.A. Turner, Modeling functional Magnetic Resonance Imaging (fMRI) experimental variables in the Ontology of Experimental Variables and Values (OoEVV), *NeuroImage*, Vol. 82, pp. 662-670, 2013.

[4] A. Gangemi, V. Presutti, Ontology Design Patterns, in *Handbook on ontologies*, pp. 221-243, 2009.

[5] A. Haller, K. Janowicz, S. Cox, M. Lefrançois, K. Taylor, D. Phuoc, J. Lieberman, R. García Castro, R. Atkinson, C. Stadler, Claus, The Modular SSN Ontology : A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation. *Semantic Web*, Vol. 10,2018.

[6] J. Hilbey, X. Aimé, J. Charlet, Représentation des connaissances médicales temporelles au moyen d'ontologies, in *33es Journées Francophones d'Ingénierie des Connaissances, IC 2022*, pp. 26-42, 2022.

[7] P. Hitzler, A. Gangemi, K. Janowicz, A.A. Krisnadhi, V. Presutti (Eds.), *Studies on the Semantic Web*, Vol. 25, 2016.

[8] A. Raboudi, M. Allanic, D. Balvay, P.-Y. Hervé, T. Viel, T. Yoganathan, A. Certain, J. Hilbey, J. Charlet, A. Durupt, P. Boutinaud, B. Eynard, B. Tavitian, The BMS-LM ontology for biomedical data reporting throughout the lifecycle of a research study : From data model to ontology, *J Biomed Inform.*, Vol. 127, 2022.

[9] L.N. Soldatova, R.D. King, An ontology of scientific experiments *Journal of The Royal Society Interface*, Vol. 3, pp. 795-803, 2006.

[10] J.A. Turner, A.R. Laird, The Cognitive Paradigm Ontology : Design and Application, *Neuroinformatics*, pp. 57-66, 2012.

Modélisation des ingrédients de remèdes issus de pharmacopées arabes médiévales dans une base de données graphe

Karim El Haff[1,2], Agnès Braud[1], Florence Le Ber[1], Véronique Pitchon[2]

[1] Université de Strasbourg, CNRS, ENGEES, ICube UMR 7357, F67000 Strasbourg
[2] Université de Strasbourg, CNRS, Archimède UMR 7044, F67000 Strasbourg
{kelhaff, agnes.braud, florence.le-ber, pitchon}@unistra.fr

Résumé

Cet article présente le travail de modélisation engagé dans le cadre d'un projet interdisciplinaire, dont le but est d'étudier les remèdes décrits dans les pharmacopées arabes médiévales. Les informations extraites d'un texte ancien, traduit en anglais, ont été représentées dans une base de données graphe. Cette étude de cas a mis en évidence plusieurs problèmes de modélisation, notamment pour la représentation d'ingrédients qui sont des sous-parties de plantes et dont l'appellation comporte des ambiguïtés. L'article détaille ces problèmes et les solutions apportées.

Mots-clés

Base de données graphe, modélisation de connaissances, textes anciens.

Abstract

This paper presents the modelling work undertaken as part of an interdisciplinary project, whose aim is to study the remedies described in medieval Arabic pharmacopoeias. Information collected from an ancient medical text, translated in English, were represented in a graph database. This study highlighted several modelling issues, including the representation of ingredients that are plant subparts, and whose name are often ambiguous. This paper details these problems and the proposed solutions.

Keywords

Graph database, knowledge modelling, ancient texts.

1 Introduction

Les textes médicaux anciens contiennent une mine de connaissances sur les maladies, les traitements et les pratiques de guérison. Ces dernières années, ces textes ont suscité un intérêt croissant [7, 16] et certains ont fait l'objet d'approches de type fouille de données [4].
Le projet PARADISE (MITI80 CNRS, 2021) s'inscrit dans cette lignée. Il a pour objectif de développer des approches informatiques pour exploiter les informations contenues dans les textes des pharmacopées arabes médiévales : il s'agit d'extraire des informations concernant principalement les remèdes associés aux maladies infectieuses, les organiser et les interroger afin de mettre en évidence puis de tester des ingrédients qui pourraient être utilisés dans la création de médicaments pouvant, en particulier, être une alternative aux antibiotiques. Le projet PARADISE regroupe pour cela des historiens, pharmaco-botanistes, biologistes et informaticiens.
Le projet se décline en plusieurs étapes. La première consiste à extraire des textes anciens (pour le moment des traductions anglaises de ces textes) les descriptions des remèdes d'intérêt pour les biologistes. Dans cette première étape, les termes utiles sont annotés et servent à l'apprentissage d'un système de reconnaissance d'entités nommées [5]. Dans la deuxième étape, les termes extraits et des informations annexes sont représentés dans une base de données, afin de permettre leur interrogation.
La majorité des remèdes extraits du corpus comporte des ingrédients à base de plantes, qui sont utilisées en tout ou partie, avec ou sans préparation ou transformation. Dans cet article, nous nous focalisons sur la modélisation de ces ingrédients, de sorte à faciliter leur interrogation. En particulier, il faut à la fois pouvoir retrouver des plantes utilisées dans différents remèdes, mais aussi les parties de plantes concernées, qui peuvent avoir un effet spécifique. Pour répondre à ces besoins, et en l'absence d'un modèle du domaine préexistant, nous avons fait le choix d'utiliser une base de données orientée graphe, pour la souplesse de représentation qu'elle offre. Néanmoins, différentes problématiques de modélisation sont apparues, que nous présentons et discutons.
L'article est organisé comme suit : après cette première partie introductive, la section 2 décrit brièvement les principes des bases de données orientées graphes, puis présente des travaux de modélisation voisins du nôtre. La section 3 présente les données et connaissances mobilisées, la section 4 décrit les différents problèmes de modélisation rencontrés, ainsi que les choix réalisés et leurs limites. La dernière section dresse quelques conclusions et perspectives.

2 Travaux connexes

2.1 Bases de données orientées graphes

Dans le domaine des bases de données, depuis la fin des années 1960, la structure de table interconnectée de la base de données relationnelle a été le modèle dominant de stockage et d'interrogation de données. Avec la croissance des données produites par les réseaux sociaux et la nécessité de trai-

ter efficacement de telles données, de nouveaux systèmes de gestion de données ont été développés. Un système de gestion de bases de données orienté graphe [1] permet de gérer des données en s'appuyant sur une représentation sous forme de graphe.

Différents travaux ont été menés autour des bases de données orientées graphes, par exemple pour adapter les algorithmes de parcours de graphes aux différentes requêtes possibles [14]. D'un point de vue applicatif, ces bases sont largement utilisées pour modéliser les réseaux sociaux. Des applications plus spécifiques ont été développées pour par exemple analyser les termes de requêtes et de leurs reformulations [12] ou pour représenter des référentiels de modèles en génie logiciel [11]. Ce type de base de données est également largement utilisé dans le domaine biomédical [17].

2.2 Modélisation de connaissances

Les bases de connaissances, ou ontologies [6], ont été développées depuis de nombreuses années dans différents domaines. Elles permettent de formaliser les connaissances d'un domaine, de les partager et de les utiliser dans des raisonnements formels. L'avantage des ontologies est aussi la possibilité de les réutiliser et de les étendre à des sous-domaines. En particulier, les ontologies développées dans les domaines végétal et médical pourraient être des supports à nos travaux de modélisation. Toutefois chaque ontologie étant développée avec un objectif spécifique, sa réutilisation nécessitera toujours un travail de modélisation important et l'appel à différentes sources de données et de connaissances, comme explicité par [3] qui présente une ontologie pour la médecine d'urgence.

Plant Ontology (PO) [18] est une ressource communautaire qui comprend des termes normalisés, des définitions et des relations décrivant les structures et les stades de développement des plantes. Cette ontologie est complétée par une base de données d'annotations provenant d'études génomiques et phénomiques. Une ontologie spécifique [9] a été construite pour la plante *Arabidopsis Thaliana* qui est utilisée comme plante modèle pour la botanique et d'autres sciences végétales. Cette ontologie vise à décrire les caractéristiques physiques, biochimiques et génétiques d'*Arabidopsis Thaliana* et les relations entre ces caractéristiques. La base de connaissances Knomana (*KNOwledge MANAgement on pesticide plants in Africa*) recense les connaissances actuelles sur les plantes utilisées comme pesticides en Afrique [15].

Ces différentes ontologies du domaine végétal décrivent les caractéristiques globales des plantes, et dans une visée (à part la base Knomana) plutôt génétique ou productive, alors que dans les remèdes issus des pharmacopées anciennes apparaissent des parties de plantes (par exemple les coques de glands ou des pépins de melon) qui peuvent avoir été transformées (grillées, séchées, etc.). C'est la modélisation de ces ingrédients que nous présentons dans la suite de l'article.

3 Présentation des données

Dans cette section nous présentons le type d'informations que nous manipulons et la base constituée.

3.1 La collecte des données initiales

Le corpus exploré est un manuscrit médical qui décrit 292 remèdes ou préparations. C'est une partie de la traduction anglaise par Oliver Kahl de l'ouvrage « Dispensatory in the Recension of the 'Aḍudī Hospital » écrit par Sābūr ibn Sahl au IXe siècle [8]. Le corpus est constitué de 36 961 *tokens* qui ont été annotés dans le but de servir à l'entraînement d'un modèle de reconnaissance d'entités nommées (*Named Entity Recognition*, NER) [5].

Les remèdes concernent principalement des maladies infectieuses, touchant différents organes du corps (poumon, peau, etc.) et générant des symptômes (toux, saignement, douleurs etc.). La description d'un remède est composée d'une liste d'ingrédients avec des quantités et une description de la préparation. Toutes les informations contenues dans les remèdes ne sont pas exploitées dans ce travail. Pour effectuer l'annotation, seules quatre étiquettes ont été utilisées : Type (pour la forme du remède), Sym (pour les symptômes), Ing (pour les ingrédients) et Org (pour les organes). Les données sont annotées dans un fichier CSV dans le format IOB2 [1] L'annotation et le nettoyage du corpus ont été effectués par le premier auteur pendant un mois et revus en profondeur par la dernière autrice, historienne experte en médecine arabe médiévale.

Ce corpus annoté a donc deux utilités : il constitue une ensemble de données d'entraînement pour la NER, dont le résultat pourra être utilisé pour l'analyse d'autres manuscrits ; il constitue également un ensemble permettant la construction d'un premier modèle de données.

3.2 L'enrichissement des données

Après l'annotation du texte à l'aide de IOB2, nous avons identifié les ingrédients uniques présents dans le texte, soit 957 ingrédients. Les termes désignant les ingrédients végétaux ont ensuite été vérifiés avec l'aide d'un botaniste pour identifier la plante concernée et y rattacher ses propriétés. Un tableur (fichier de type CSV) a été créé et complété avec un ensemble d'informations obtenu à partir des ressources The World Flora Online [2] pour la botanique, CHEMnetBASE [3] et Reaxys [4] pour les molécules naturelles.

Le tableau 1 en présente un extrait : il contient le nom de l'ingrédient tel qu'il est écrit dans le livre (exemple : absinthe leaves), le nom vernaculaire de la plante ainsi que des synonymes courants (absinthe, wormwood [...]), la partie de la plante utilisée (leaf), le nom scientifique de la plante (*Artemisia absinthium*), les synonymes scientifiques de son nom (*Absinthium bipedale* [...]), la famille de la plante (*Asteraceae*), son origine géographique (Afghanistan, Albania,

1. IOB2, abréviation de « inside, outside, beginning » est un format de marquage commun en traitement automatique des langues [13].
2. `http://www.worldfloraonline.org/`
3. `https://dnp.chemnetbase.com/`
4. `https://www.reaxys.com/`

Ingrédient	Nom vernaculaire	Partie de la plante	Nom Scientifique	Synonymes	Famille	Origine géographique	Molécules actives	Toxicité
absinthe leaves	absinthe, wormwood [...]	leaf-_absinthe	*Artemisia absinthium*	*Absinthium bipedale* [...]	*Asteraceae*	Afghanistan, Albania, Algeria [...]	polyphenol, monoterpene	whole plant
absinthe sap	absinthe, wormwood [...]	sap_absinthe	*Artemisia absinthium*	*Absinthium bipedale* [...]	*Asteraceae*	Afghanistan, Albania, Algeria [...]	polyphenol, monoterpene	whole plant
citrons from Susa	citron tree from susa	fruit_citron-tree	*Citrus Medica*	*Citrus acida* [...]	*Rutaceae*	Assam, Bangladesh, East Himalaya [...]	polyphenol, coumarin, flavonoid, terpene	none
peels of celery roots	celery	peel_root-_celery	*Apium Graveolens*	*Apium australe var. latisectum* [...]	*Apiaceae*	Afghanistan, Albania, Algeria, [...]	glycosid, polyphenol, furocoumarin	whole plant low

TABLE 1 – Extrait du tableau de données : les termes issus du texte sont complétés par des informations sur les plantes correspondantes

Algeria [...]), les molécules actives trouvées dans la plante (polyphenol, monoterpene), et sa toxicité (whole plant).

3.3 Le modèle des données

Nous avons développé un modèle pour représenter dans une base de données orientée graphe les relations complexes entre les remèdes, les ingrédients, les plantes et leurs parties, ainsi que les transformations d'ingrédients. Une représentation de ce modèle est présenté en figure 1. Nous utilisons l'outil Neo4j [5], qui offre une interface d'interrogation et de visualisation des données.

Le modèle est structuré autour de la relation `Contains` (contient, 1700 instances) entre les nœuds `Remedy` (remède, 287 inst.) et `Ingredient` (ingrédient, 715 inst.). Cette relation est dotée de deux attributs : le nom original de l'ingrédient dans le manuscrit, ainsi qu'une origine géographique spécifiée dans le nom le cas échéant, comme par exemple *Antioch* pour l'ingrédient *Antioch scammony* (scammonée d'Antioche). Les ingrédients peuvent être une plante entière ou une partie de plante : le nœud `Ingredient` est alors relié par la relation `PartOf` (partie de, 259 inst.) à un autre nœud de type `Ingredient` (plante ou partie de plante). Un nœud `Ingredient` est également relié à un nœud `Taxon` (311 inst.), grâce à la relation `HasTaxon` (a comme taxon, 503 inst.). Ce nœud possède 5 attributs : le nom scientifique de la plante (espèce ou genre), la liste des synonymes scientifiques, la liste des origines géographiques possibles, la liste des principes actifs et la toxicité. De plus, chaque nœud `Taxon` est relié par `FromFamily` (de la famille, 312 inst.) à un nœud `Family` (famille, 114 inst.), qui représente une famille de plantes. Enfin, notre modèle prend en compte les transformations d'ingrédients dans les remèdes. La relation `ContainsTransformed` (contient transformé, 560 inst.) relie les remèdes aux ingrédients transformés (nœud `TransformedIngredient`, 259 inst.) et possède les mêmes attributs que la relation `Contains`. La relation `IsFrom` (provient de, 258 inst.) permet de relier les ingrédients transformés à l'ingrédient d'origine, avec l'attribut `transformation_type` qui permet de catégoriser la transformation.

5. `https://neo4j.com/fr/`

4 Questions de modélisation

Définir le modèle de données pour enregistrer les remèdes issus du corpus étudié dans la BD graphe a soulevé plusieurs questions que nous développons ci-dessous.

4.1 Les ingrédients sous-parties de plantes

La première question portait sur la façon de modéliser les parties de plantes utilisées comme ingrédients dans les remèdes. En effet, nombre d'ingrédients utilisés dans les remèdes ne sont des parties de plantes, telles que des graines, des fruits, des racines, des feuilles, etc.

Dans le tableau 1, la colonne "Partie de la plante" précise la partie correspondant à chaque ingrédient mentionné dans un remède. Par exemple, si l'ingrédient original est *apple seeds* (pépins de pomme), la colonne "Partie de la plante" contiendra "seed_fruit_apple tree". Pour formaliser ces informations, il est intéressant de décomposer la chaîne menant d'une sous-partie à la partie entière de la plante. Par exemple, pour l'ingrédient "apple seeds", la chaîne va de "Seed : seed_fruit_apple tree" à la plante "Plant : apple tree", en passant par "Fruit : fruit_apple tree". On voit ici trois catégories de parties de plantes, la graine, le fruit et l'arbre (plante entière).

Chaque partie d'une plante a été modélisée comme un nœud dans un arbre hiérarchique. La racine de l'arbre est le nœud "plante entière", qui est lié au nom scientifique de la plante dans notre modèle.

Les sous-parties de plantes ont été regroupées en 21 types, sous-types du nœud `Ingredient` : `Fruit` (fruit), `Pulp of fruit` (pulpe de fruit), `Inner skin` (peau intérieure : couche interne d'un fruit, souvent mince et fibreuse), `Peel` (pelure : couche externe du fruit), `Shell` (coquille : couche dure et extérieure de certains fruits), `Seed` (graine), `Pulp of seed` (pulpe de graine), `Seed core` (cœur de graine), `Seed vessel` (enveloppe de la graine), `Stem` (tige), `Leaf` (feuille), `Twig` (rameau), `Stalk` (pédoncule : tige qui porte la fleur puis le fruit), `Flower` (fleur), `Flower buds` (bourgeons floraux), `Root` (racine), `Root peel` (pelure de racine : couche externe de la racine), `Bark` (écorce), `Mucilage` (mucilage : substance visqueuse et épaisse produite par certaines plantes), `Sap` (sève), `Gall` (galle : excroissance anormale de la plante causée par une infection bactérienne

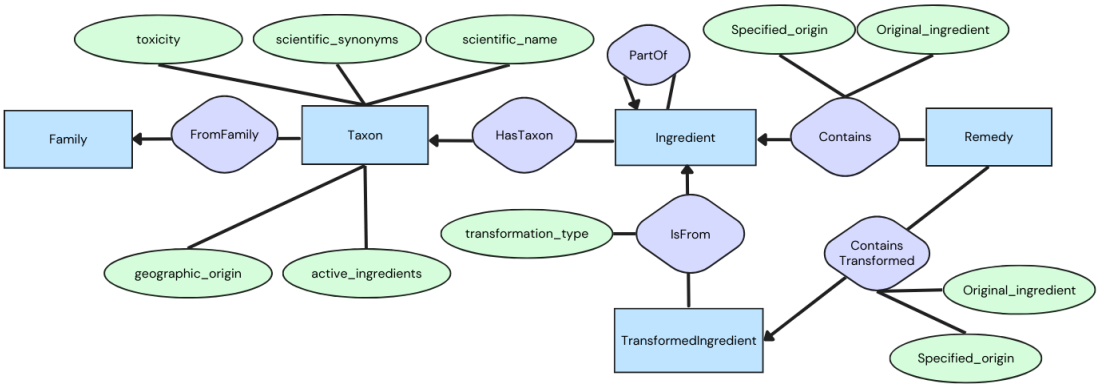


FIGURE 1 – Le modèle de la base de données, les nœuds sont symbolisés par des rectangles, les relations par des losanges et les attributs par des ellipses

ou fongique).

Finalement chaque ingrédient est décomposé selon la chaîne des sous-parties qui le constituent. Cette décomposition a été faite manuellement et un script python permet ensuite de remplir la base de données selon le modèle de la figure 1 en s'appuyant sur la hiérarchie des types de sous-parties de plantes. Cela a permis de modéliser les parties de plantes de manière plus précise et de relier les remèdes aux plantes dont leurs ingrédients sont issus pour faciliter la recherche d'informations. Par exemple, la figure 2 montre le graphe reliant l'ingrédient *citron* aux remèdes 41 et 252, et l'ingrédient *citron peels* au remède 252.

4.2 Les ingrédients transformés

Une deuxième question concerne la représentation des ingrédients transformés, dont la la composition chimique et les propriétés médicinales ont ainsi été modifiés par rapport à l'ingrédient original. Pour représenter cette information dans la base de données graphe, on utilise un type de nœud appelé `TransformedIngredient`, lié à l'ingrédient non transformé par une relation `IsFrom`, qui a comme attribut le type de transformation (séchage, broyage, etc.). Le nœud `TransformedIngredient` est ensuite lié au remède avec la relation `ContainsTransformed`. Cela permet de représenter l'ingrédient végétal original et l'ingrédient transformé, ainsi que les détails du processus de transformation.

Cette approche présente plusieurs avantages. Premièrement, en reliant l'ingrédient végétal original et l'ingrédient transformé qui en résulte, elle permet de retrouver par une seule requête (exploitant la structure de graphe) les remèdes contenant ce végétal, transformé ou non. Deuxièmement, en représentant explicitement l'ingrédient transformé, elle permet aux biologistes d'intégrer les effets spécifiques du processus de transformation sur la composition chimique de l'ingrédient et son effet potentiel. Enfin, d'un point de vue formel, cette représentation permet d'inclure des ingrédients dont la description comporte une négation ou une soustraction, ce qui n'est pas représentable directement par des graphes. Par exemple, l'ingrédient *seedless barberries* (baies d'épine-vinette sans pépins) est difficile à représenter sauf si nous considérons *l'épépinage* comme une transformation (voir figure 3).

4.3 Questions ouvertes

4.3.1 Problèmes sémantiques

Les données ont été extraites d'un texte en anglais, traduit d'un texte arabe ancien, et à ce titre, elles sont entachées d'ambiguïtés sémantiques, que nous ne pouvons lever directement et qui induisent des imprécisions dans la représentation. Nous en donnons deux exemples ci-dessous. Dans les deux cas, pour lever les ambiguïtés, il sera nécessaire de revenir au texte originel, en langue arabe, avec l'aide de spécialistes du lexique de la médecine arabe médiévale et avec l'appui de botanistes.

Appellation courante et nom scientifique. L'un des problèmes que nous avons rencontrés lors de la modélisation est celui des noms de plantes, ou de parties de plantes, ambigus. Dans de nombreux cas, les noms des plantes apparaissant dans le manuscrit traduit ne peuvent être mis en correspondance univoque avec une dénomination scientifique, car ce sont des appellations courantes, qui peuvent correspondre à différentes espèces.

Par exemple, le terme *acorn* (gland), peut désigner en langage courant soit le fruit d'une espèce *Quercus sp* (*oak*), soit celui d'une espèce *Lithocarpus sp* (*stone oak*). Sans une référence plus spécifique à l'espèce en question, il est impossible de catégoriser les données avec précision. Pour ce cas, nous avons donc choisi d'assigner le double nom scientifique *Quercus sp ;Lithocarpus sp*.

Indication d'origine dans une appellation courante. Les désignations des ingrédients peuvent porter une information géographique, comme par exemple *Syrian apples* ou *Chinese rhubarb*. C'est le cas pour 50 ingrédients dans la base de données actuelle. Nous avons choisi de représenter cette information dans un attribut de la relation `Contains` reliant le remède et l'ingrédient. De cette manière, nous cherchons à généraliser les nœuds qui désignent les ingrédients ayant un même nom scientifique pour faciliter les requêtes dans un premier temps, tout en conservant l'information sur l'origine géographique. Toutefois, la sémantique

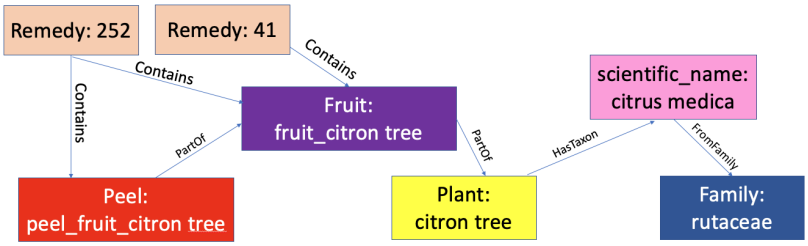


FIGURE 2 – Graphe reliant des remèdes contenant des ingrédients correspondant à des sous-parties du citronnier : le remède 252 a pour ingrédients à la fois le fruit et des pelures (zestes) du fruit.

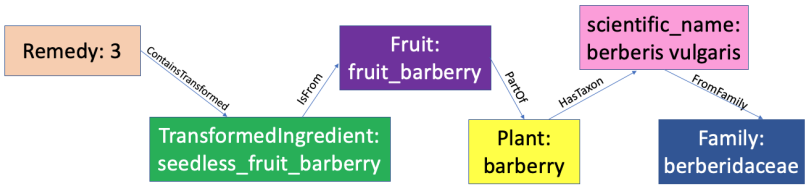


FIGURE 3 – Graphe d'un remède contenant des baies d'épine-vinette épépinées (*seedless barberries*)

portée par cette information peut être variable : elle peut effectivement désigner une origine géographique spécifique, choisie de préférence (le vin de Falerne a certaines propriétés, les dattes de Hairūn sont réputées) ou bien relever d'une appellation courante. Par exemple, le terme *Greek absinthe* pourrait suggérer qu'il s'agit d'une absinthe originaire de Grèce, mais nous ne pouvons pas en être certains sans recherches supplémentaires. De fait, distinguer une origine spécifique d'un terme utilisé couramment est important car une même plante peut avoir des propriétés différentes selon la région où elle est cultivée [10].

4.3.2 Les doubles ingrédients

Certains termes étiquetés comme ingrédients, extraits du corpus, recouvrent en fait plusieurs ingrédients. Par exemple, l'ingrédient *seed vessels and flowers of the pomegranate* (enveloppes de graines et fleurs de grenade) est une combinaison de deux parties différentes du grenadier. L'ingrédient *celery- and fennel-water* (l'eau de céleri et de fenouil) est un exemple plus complexe, où intervient une transformation de deux plantes distinctes.

Dans la base de données actuelle, il y a 28 cas de tels ingrédients combinés. Pour faciliter l'interrogation de la base – par exemple retrouver tous les remèdes contenant des fleurs de grenadier – il est intéressant de décomposer ces ingrédients combinés en les ingrédients qui les composent.

Cependant, les désignations d'ingrédients combinés peuvent s'interpréter de deux façons : soit la combinaison est liée à la syntaxe du langage naturel qui a tendance à éviter la répétition de l'objet – graines et fleurs de grenade – facile à décomposer par un traitement syntaxique à base de règles (*A et B de C* devient *A de C et B de C* : graines de grenade et fleurs de grenade) ; soit elle est liée à la fabrication, par transformation, de l'ingrédient – l'eau de céleri et de fenouil : dans ce cas, savoir si les deux ingrédients sont transformés (par infusion ou décoction) ensemble ou séparément nécessite une expertise du domaine ainsi que le retour au texte dans sa langue originale.

Pour garder la double information, de la combinaison des ingrédients et de leur individualité, le modèle de la base pourrait être modifié en introduisant un type de nœud `CombinedIngredient`, lié aux nœuds `Ingredient` qui le composent. Cela permettrait de saisir des informations plus détaillées tout en maintenant le lien entre la source originale et les données modélisées.

5 Conclusion et perspectives

Le travail présenté ici s'inscrit dans un projet pluridisciplinaire visant à exploiter les pharmacopées anciennes dans le but de créer de nouveaux médicaments, en particulier en alternative aux antibiotiques. Nous avons dans un premier temps travaillé sur un texte arabe médiéval, traduit en anglais, dont nous avons extrait les descriptions des remèdes, comprenant les symptômes et les organes traités ainsi que les ingrédients qui les composent.

Nous avons choisi de stocker ces éléments dans une base de données orientée graphe. Nous rendons compte dans cet article des différents problèmes de modélisation que nous avons rencontrés et traités comme la prise en compte des sous-parties des plantes et des ingrédients transformées, ou qui nécessitent un approfondissement tel que les ambiguïtés sémantiques et syntaxiques.

Le cadre offert par le modèle de graphe s'est révélé pertinent à la fois pour faciliter la représentation des informations complexes que nous manipulons et l'interrogation des données, mais aussi pour visualiser les relations entre les différents éléments des remèdes extraits du manuscrit. Cette première base de données sur les remèdes anciens suscite beaucoup d'intérêt de la part des biologistes, mais nécessite d'être complétée pour qu'ils puissent l'utiliser dans leur recherche d'ingrédients utiles à la création de médicaments.

Le travail réalisé ouvre de nouvelles perspectives pour approfondir l'exploration des connaissances contenues dans

les manuscrits anciens de médecine arabe. Dans un premier temps, il s'agit de développer des modes d'interrogation de la base de données permettant de mettre en évidence des connaissances pertinentes en utilisant l'analyse de concepts formels, comme suggéré par Braud et *al.* [2]. Parallèlement, la base de données sera complétée à partir d'autres textes, dont l'annotation est en cours. Enfin, l'analyse des remèdes pourra être enrichie par une représentation explicite des différentes taxonomies utilisées, telles que les plantes, les parties de plantes, ainsi que les transformations des ingrédients. Ces taxonomies devront être reliées à des ressources externes afin de rendre ces données réutilisables et interopérables. Ainsi complétée, la base de données pourra alors être publiée.

Remerciements

Cette étude a été financée par le CNRS dans le cadre de l'appel à projets MITI80 (2021). Nous remercions tous ceux qui ont contribué à la réalisation de ce travail : nos collègues biologistes, pharmacologues, ainsi que Anthony Masiala (Master 1 en Botanique de l'Université de Strasbourg) qui a collecté les connaissances sur les plantes.

Références

[1] Amitabha Bhattacharyya and Durgapada Chakravarty. Graph database : A survey. In *2020 International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, pages 1–8, 2020.

[2] Agnès Braud, Xavier Dolques, Pierre Fechter, Nicolas Lachiche, Florence Le Ber, and Veronique Pitchon. Analyzing the composition of remedies in ancient pharmacopeias with FCA. In *RealDataFCA'2021, ICFCA Workshop, Strasbourg, France*, CEUR Workshop Proc. 3151, pages 28–35, 2021.

[3] Jean Charlet, Gunnar Declerck, Ferdinand Dhombres, Pierre Gayet, Patrick Miroux, and Pierre-Yves Vandenbussche. Construire une ontologie médicale pour la recherche d'information : problématiques terminologiques et de modélisation. In *23es journées francophones d'Ingénierie des connaissances, Paris, France*, pages 33–48, 2012.

[4] Erin Connelly, Charo I. del Genio, and Freya Harrison. Data mining a medieval medical text reveals patterns in ingredient choice that reflect biological activity against infectious agents. *mBio*, 11(1), 2020.

[5] Karim El Haff, Wissam Antoun, Florence Le Ber, and Véronique Pitchon. Reconnaissance des entités nommées pour l'analyse des pharmacopées médiévales. In *EGC 2023 - Extraction et Gestion des Connaissances, Lyon, France*, volume RNTI-E-39, pages 329–336, 2023.

[6] T. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, 1993.

[7] Freya Harrison, Aled E. L. Roberts, Rebecca Gabrilska, Kendra P. Rumbaugh, Christina Lee, and Stephen P. Diggle. A 1,000-Year-Old Antimicrobial Remedy with Antistaphylococcal Activity. *mBio*, 6(4) :e01129–15, 2015.

[8] Oliver Kahl. *Sābūr Ibn Sahl's Dispensatory in the Recension of the 'Aḍudī Hospital*. BRILL, 2009. Arabic edition and English translation.

[9] Sara Hosseinzadeh Kassani and Peyman Hosseinzadeh Kassani. Building an Ontology for the Domain of Plant Science using protégé, 2018. arXiv :1810.04606.

[10] Wei Liu, Dongxue Yin, Na Li, Xiaogai Hou, Dongmei Wang, Dengwu Li, and Jianjun Liu. Influence of Environmental Factors on the Active Substance Production and Antioxidant Activity in Potentilla fruticosa L. and Its Quality Assessment. *Scientific Reports*, 6(1) :28591, 2016.

[11] Thierry Millan. Utilisation des bases de données orientées graphe comme référentiels de modèles. *Revue des Sciences et Technologies de l'Information - Série TSI*, 35(6) :695–719, 2017.

[12] Josiane Mothe and Sagun Pai. Mise en œuvre d'une base de données graphe pour l'analyse des logs de requêtes en recherche d'information. In *14eme Conference francophone en Recherche d'Information et Applications (CORIA 2017), Marseille, France*, pages pp. 43–58, 2017.

[13] Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040, 1995.

[14] Marko A. Rodriguez and Peter Neubauer. The Graph Traversal Pattern, 2010. arXiv :1004.1001 [cs].

[15] Pierre J. Silvie, Pierre Martin, Marianne Huchard, Priscilla Keip, Alain Gutierrez, and Samira Sarter. Prototyping a Knowledge-Based System to Identify Botanical Extracts for Plant Health in Sub-Saharan Africa. *Plants*, 10(5), 2021.

[16] Some, Borlli Michel Jonas, Georgeta Bordea, Frantz Thiessard, and Gayo Diallo. Enabling West African Herbal-Based Traditional Medicine Digitizing : The WATRIMed Knowledge Graph. In *MEDINFO 2019 : Health and Wellbeing e-Networks for All*, pages 1548–1549. IOS Press, 2019.

[17] Santiago Timón-Reina, Mariano Rincón, and Rafael Martínez-Tomás. An overview of graph databases and their applications in the biomedical domain. *Database*, 2021 :1–22, 2021.

[18] Ramona L. Walls, Laurel Cooper, Justin Elser, Maria Alejandra Gandolfo, Christopher J. Mungall, Barry Smith, Dennis W. Stevenson, and Pankaj Jaiswal. The plant ontology facilitates comparisons of plant development stages across species. *Frontiers in Plant Science*, 10, 2019.

Besoins ontologiques pour la transformation des aliments

D. Dooley[1], M. Weber[2], L. Ibanescu[3], M. Lange[4], L. Chan[5], L. Soldatova[6], C. Yang[7], R. Warren[8], C. Shimizu[9], H. McGinty[10], W. Hsiao[1,11]

[1] Centre for Infectious Disease Genomics and One Health, Simon Fraser University, Burnaby, BC, Canada
[2] INRAE, UR BIA, Biopolymères Interactions Assemblages, Nantes, France
[3] Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay, Palaiseau, France
[4] International Center for Food Ontology Operability Data and Semantics, Davis, California, USA
[5] College of Public Health and Human Sciences, Oregon State University, Corvallis, OR, USA
[6] Goldsmiths College, the University of London, UK
[7] Department of Food Technology, Safety and Health, Ghent University, Ghent, Belgium
[8] Glengarry Agriculture and Forestry, Guelph, On, Canada
[9] Kansas State University, Manhattan, KS, USA
[10] Department of Chemistry and Biochemistry, Ohio University, Athens, Ohio, USA
[11] Faculty of Health Sciences, Simon Fraser University, Burnaby, BC, Canada

liliana.ibanescu@agroparistech.fr

Résumé

La nourriture est souvent appréciée pour ses aspects sensuels, festifs et pour sa contribution à la santé, mais derrière se cachent de nombreux autres processus de production agricole, de distribution, de fabrication et des processus physiologiques qui peuvent maintenir ou miner la santé de la population. La complexité de ces processus est évidente à la fois dans la préparation quotidienne des repas et dans la fabrication industrielle, l'emballage et le stockage des aliments. Dans ce voyage de la ferme à la fourchette il n'existe pas encore d'ontologie pour couvrir toutes les étapes et il semble nécessaire de construire une telle vision en réutilisant des ontologies existantes pour des sous-domaines, organisés par des experts. Le défi est que cette fusion soit, par analogie, une seule langue, plutôt que des noms et des verbes d'une douzaine de dialectes. Ce travail se concentre sur les composants de l'ontologie – propriétés des objets et des données et annotations - nécessaires pour modéliser les processus alimentaires ou, plus général, la modélisation de processus, dans le contexte de l'Open Biological and Biomedical Ontology Foundry et des ontologies congruentes.

Mots-clés

Ontologie, transformation des aliments, recette, modélisation des processus, OBO Foundry

Abstract

People often value the sensual, celebratory, and health aspects of food, but behind this experience exists many other value-laden agricultural production, distribution, manufacturing, and physiological processes that support or undermine a healthy population. The complexity of such processes is evident in both every-day food preparation of recipes and in industrial food manufacturing, packaging and storage. An integrated ontology landscape does not yet exist to cover all the entities at work in this farm to fork journey. It seems necessary to construct such a vision by reusing expert-curated fit-to-purpose ontology subdomains. The challenge is to make this merger be, by analogy, one language, rather than nouns and verbs from a dozen or more dialects. This work focuses on the ontology components – object and data properties and annotations – needed to model food processes or more general process modelling within the context of the Open Biological and Biomedical Ontology Foundry and congruent ontologies.

Keywords

Ontology, food processing, recipe, process modelling, OBO Foundry

1 Introduction

La transformation des aliments a, comme Janus, deux visages opposés, offrant d'une part des progrès significatifs qui rendent les aliments plus résistants à la détérioration, et donc plus faciles à conserver, plus nutritifs, plus savoureux, plus pratiques et souvent peu coûteux [17]. D'autre part, nombre de ces mêmes caractéristiques ont encouragé la surconsommation, les régimes alimentaires malsains et des produits alimentaires qui perturbent la dynamique du microbiome intestinal humain et la santé en général [13]. La recherche en génie des procédés est apparue dans les années 1950 pour comprendre les implications économiques et de santé humaine d'un système de fabrication alimentaire de plus en plus industrialisé et mondialisé, en prenant en compte la transformation, la production, la manutention,

le stockage, la conservation, le contrôle, l'emballage et la distribution des produits alimentaires. L'ingénierie alimentaire est un domaine multidisciplinaire qui combine la microbiologie, les sciences physiques appliquées et le génie chimique pour comprendre et concevoir des produits et des opérations dans les industries de l'alimentation.

Les citoyens sont de plus en plus préoccupés par les questions de santé et l'impact sur l'environnement, et ils recherchent également des aliments savoureux, naturels et pratiques. Aujourd'hui, l'ingénierie alimentaire s'appuie sur les données que les systèmes d'information apportent pour répondre aux exigences de santé grâce à une nutrition de précision s'appuyant sur des connaissances scientifiques afin de répondre aux besoins nutritionnels des individus en fonction de différents facteurs, tels que l'âge, l'état de santé ou une pathologie, les facteurs liés à l'activité ou au mode de vie et d'éviter des aliments ou ingrédients qui produisent des réponses allergiques. Les fabricants de produits alimentaires trouvent également une valeur ajoutée à travers l'adaptation de leurs produits aux demandes des consommateurs qui basent aujourd'hui leurs choix alimentaires sur certains critères éthiques (par exemple, le bien-être des animaux), qui vont au-delà des courants de consommation particuliers que sont le casher, le halal et le végétarisme.

La prise de décision complexe et interdépendante au sein d'un paysage d'acteurs multiples —de la production, de la distribution, de la consommation et de la réglementation — nécessite un registre normalisé des informations collectées tout au long des cycles de vie des produits alimentaires. Les systèmes de jumeaux numériques construits à partir de normes d'identification des produits comme le GS1 Digital Link, combinés à une infrastructure de graphe de connaissances comme Origin Trail [21], fournissent aux parties prenantes un mécanisme de saisie des données contextuelles sur les produits, de la ferme à l'assiette, mais ce système sera peu utilisable s'il ne s'agit que d'un potpourri de langages restant chacun cloisonné et propre à son contributeur. Les participants à cette chaîne d'information bénéficieront de la mise au point d'une ontologie — un langage commun, standardisé, exploitable par les machines — qui recouvre la taxonomie et l'anatomie des organismes sources des matières premières, les informations sur le contexte de la production agricole et du transport, ainsi que les méthodes de transformation et de préparation des aliments.

L'un des principaux objectifs de l'analyse présentée dans notre article [11] est de formuler des recommandations pour un cadre générique de modélisation des processus qui soit basé sur l'initiative et les recommandations d'OBO Foundry (Open Biological and Biomedical Ontology Foundry) [15]. Notre objectif principal est de passer en revue des ontologies OWL qui possèdent des classes et des propriétés nécessaires à un modèle de processus OBO. Cette analyse nous a permis, d'une part, d'identifier les entités nécessaires pour modéliser la transformation des aliments et, de manière plus générale, les protocoles de laboratoire ou les recettes de fabrication, et, d'autre part, de comparer les différentes manières de représenter les dépendances temporelles, les enchaînements, les entrées et les sorties des processus, les participants ou agents impliqués dans un processus ou une partie du processus. Nous avons examiné l'ontologie Exact2 (EXperimental ACTions) [23], l'ontologie PO2 (Process and Observation Ontology) [10], le modèle de processus planifié OBI (Ontology for Biomedical Investigations)[9], qui présentent divers degrés de compatibilité avec OBO, ainsi que l'ontologie PROV-O [20], l'ontologie OWL Time [18] et l'ontologie SOSA (Sensor, Observation, Sample, and Actuator) [16], dont l'adoption a connu un essor considérable. Enfin, nous avons analysé le modèle de recette de Schema.org, qui comporte un certain nombre d'éléments liés aux processus.

Après cet examen, nous constatons que certaines propriétés objets peuvent être réutilisées telles quelles (comme OWL-Time "hasTime"), tandis que d'autres nécessiteraient la création d'entités OBO équivalentes, pour respecter les principes OBO. Ces entités peuvent ensuite être ajoutées à l'ontologie alimentaire FoodOn [12], qui est un pivot au sein de l'initiative OBO permettant de relier les caractéristiques des produits alimentaires et les aspects liés à la transformation des aliments, ce qui est essentiel pour la traçabilité des aliments et d'autres applications de la ferme à la fourchette. Enfin nous proposons un modèle pour représenter un processus de transformation alimentaire se basant sur la comparaison avec les autres ontologies et nous concluons en présentant un cas d'utilisation sous la forme d'une simple recette de purée de carotte ou "carottes bouillies". Nous décomposons la recette en étapes, avec des informations qui pourraient satisfaire le consommateur qui a simplement besoin d'instructions pour réaliser la recette à la maison, l'industriel qui recherche une formulation pour cibler des caractéristiques précises pour le produit final et traduire cette formulation en processus industriel. Le modèle de recette que nous proposons est déjà utilisé dans un article récent [24] qui propose un nouveau modèle de conception ontologique pour étudier les substitutions possibles d'ingrédients au sein d'une recette [4].

Dans cet article en français nous avons fait le choix de ne présenter que le cadre général et le modèle de FoodOn detaillé dans article [11].

2 Modélisation de processus basée sur une ontologie

2.1 OBO Foundry : le cadre

Nous proposons de faire la modélisation des processus dans le cadre d'OBO Foundry parce que nous sommes favorables à une stratégie qui consiste à avoir un ensemble minimal de propriétés de données, un ensemble réduit de propriétés d'objets et à mettre davantage l'accent sur la définition des classes d'entités reliées par une propriété d'objet. Une deuxième motivation pour utiliser OBO - bien qu'elle ne soit pas propre à son cadre - est d'encourager la normalisation en réduisant le nombre de termes sémantiquement dupliqués dans les ontologies membres, promouvant ainsi une compréhension encyclopédique. Les ontologies membres

de OBO sont encouragées à réutiliser une grammaire de base des relations, fournie principalement par l'ontologie des relation RO (Relation Ontology) [1]. Cela permet de respecter le deuxième engagement de la communauté OBO, à savoir que les ontologies membres soient logiquement compatibles les unes avec les autres. En ce qui concerne la cohérence logique, OBO a tacitement encouragé le raisonnement sur la structure logique d'une ontologie donnée fusionnée avec l'ontologie formelle de base (BFO) [8] pour mettre en évidence les incohérences internes. Plus récemment, un autre point de départ de la compatibilité est le COB (OBO Core Ontology for Biology and Biomedicine) [22], une ontologie en cours de développement qui comprend un ensemble réduit de classes et de relations de niveau supérieur couramment utilisées - entité matérielle, processus, caractéristique et information - combinées en une seule ressource avec pour objectif principal de prendre en charge les contraintes de domaine et de co-domaine des relations de RO.

Pour s'intégrer dans la communauté OBO, une ontologie doit mettre en œuvre les principes OBO. Si des composants utiles ne sont pas réutilisables tels quels dans l'OBO, ils doivent être remplacés par des termes comparables répondant aux critères de l'OBO [3], dont voici quelques exemples :

- Gestion des URL permanentes (PURL) : Chaque terme de l'ontologie se voit attribuer une URL et est rattaché à un service qui renvoie des informations lisibles par l'homme et par l'ordinateur sur le terme. L'URL du terme est censée exister en permanence ; il existe un système de référence pour la dépréciation et le remplacement des termes qui facilite les mises à jour de la base de données en fonction de l'évolution des ontologies.
- Normes de curation : Les termes sont désignés au singulier, en anglais et en minuscules, sauf pour les noms propres. Chaque terme est accompagné d'une définition aristotélicienne qui fait référence à la classe d'origine et le différencie de ses frères et sœurs. Les auteurs des termes et les sources des définitions sont mentionnés.
- Axiomatisation : Les termes sont, dans une certaine mesure, logiquement reliés par des relations avec d'autres entités.
- Collaboration : Une ontologie importe le terme d'une autre ontologie plutôt que de reproduire la même sémantique dans l'un de ses propres termes conformément au principe de l'information minimale pour référencer un terme d'ontologie externe (MIREOT) [14]. Au sein d'OBO Foundry, des ontologies "de référence" définissent des domaines qui sont les ressources de référence pour les autres ontologies qui en ont besoin, donc idéalement une ontologie pour la taxonomie, une pour la chimie, une pour l'anatomie, etc., mais il faut admettre que ce niveau de qualité est encore loin d'être atteint.

2.2 Capacités attendues du modèle de processus

Il existe différents types de processus physiques, chimiques, biologiques ou de transformation des données, ainsi que des processus intentionnels qui les exploitent. Nous pouvons modéliser des processus non planifiés présents dans le monde physique, qui peuvent ensuite être exploités par des processus planifiés impliquant un ou plusieurs objectifs. Par exemple, le mûrissement des fruits, un processus biologique, peut être manipulé par un mûrissement artificiel et/ou une récolte et un transport planifiés qui, ensemble, répondent à l'objectif de livrer des fruits mûrs aux clients. Les processus planifiés ont un plan ou un protocole détaillé à réaliser, et agissent sur des entrées et des sorties qui sont des entités matérielles ou des données. Un processus planifié peut impliquer des transformations intrinsèques ou extrinsèques :

- Transformer la composition d'une entité par des processus de mécaniques, chimiques, biologiques ou autres processus de production physiques.
- Caractériser (générer des informations sur) une entité en utilisant des processus d'observation qui peuvent être invasifs ou non invasifs.
- Modifier le contexte relatif ou les relations extrinsèques d'une entité, comme le transport d'objets d'un endroit à un autre.
- Affecter passivement une entité, par exemple un objet stocké dans un objectif de conservation.

Les processus peuvent être organisés de manière linéaire ou combinés dans des réseaux plus complexes qui se comportent collectivement comme des opérations par lots ou continues. Ils peuvent être limités par des dépendances et la disponibilité des intrants, et nécessiter diverses ressources et exécutants tels que des appareils et des personnes. Ils peuvent avoir des durées minimales nécessaires, des taux de changement, des sous-processus et des effets secondaires.

L'un des objectifs d'une ontologie de processus générale est de couvrir des niches d'ontologie de processus plus spécifiques - de la traçabilité des aliments, qui nécessite une granularité grossière du processus - récolte, stockage, transport, division, combinaison - à une modélisation plus spécifique, telle que la manière d'élaborer une recette. Nous cherchons à répondre à la question suivante : les mêmes relations génériques du modèle de processus peuvent-elles couvrir la modélisation de niches spécifiques, de sorte que les relations dans ces niches se révèlent équivalentes aux relations génériques ?

2.3 Objectifs du processus

L'objectif d'un processus peut être exprimé simplement en se référant à une entité de sortie, un produit désiré. Divers procédés peuvent être connus pour obtenir un tel résultat, mais le choix de l'un d'entre eux peut être limité par les dispositifs ou les intrants disponibles, ou par des contraintes de temps, ou par un manque d'information (par exemple, en l'absence de protocole précis). Une modélisation plus générique est obtenue en décrivant ce qu'est l'entité de sortie de telle sorte que le processus puisse être reconnu comme un moyen d'atteindre cet objectif.

Les objectifs du processus peuvent également être exprimés en faisant référence aux capacités fonctionnelles des appareils. Dans BFO, une fonction est présente dans un dis-

positif ou un matériel et un processus "réalise" ou effectue la transformation qu'une fonction caractérise. Un protocole ou plan contenant des instructions détaillées pas-à-pas peut être impliqué—il "exécute" le processus.

2.3.1 Variations de la modularité des processus

Il existe quelques types d'approches pour modéliser des processus et les capacités qu'un modèle basé sur une ontologie peut idéalement satisfaire, allant de modèles holistiques permettant d'appréhender à la fois le comportement autonome d'agents libres de se déplacer dans un environnement, recherchant ou attendant activement des informations contextuelles auxquelles ils sont censés réagir, que des modèles limités à la description des étapes d'un flux d'opérations, implicitement contrôlés par une couche plus abstraite. Les modèles de processus orientés vers l'autonomie peuvent exprimer les conditions dans lesquelles un processus est activé - un ensemble de critères concernant le contexte environnemental requis, les matériaux d'entrée, l'énergie, les contraintes de temps, le(s) dispositif(s) et les opérateurs. Les modèles de processus orientés vers le contrôle peuvent comporter une couche parallèle de processus de contrôle qui fournissent des signaux d'entrée tels que "démarrage", "pause" et "arrêt", parallèlement aux conditions d'activation orientées vers l'agent. Par exemple, la Data Documentation Initiative (DDI) [6] et sa variante récente DDI Cross-Domain Integration (DDI CDI) [2] fournissent un modèle de processus qui présente cette approche basée sur la spécification de provenance PROV du W3C, dont PROV-O fait partie [5].

Les deux types de modèles peuvent être connectés de manière « plug and play » pour créer des dépendances de processus et une transformation globale du matériau/phénotype d'un état initial à un état final. Les moteurs de règles mettent en œuvre l'approche autonome en surveillant l'état de l'environnement. Un processus autonome commence à ressembler à un processus orienté vers le contrôle quand son environnement est réduit à un ensemble étroit de stimuli/entrées (les entrées qui provoquent des réactions deviennent plus évidentes).

Les spécifications du flux d'opérations et leurs cadres informatiques permettant la définition d'un réseau de processus constituent l'ossature du traitement des données et de la reproductibilité expérimentale. La configuration du flux de d'opérations est souvent un processus de configuration manuelle, mais de nombreux systèmes tels que CyVerse [19] et Galaxy [7] fournissent des interfaces utilisateur pour le développement du flux d'opérations qui indiquent les entrées requises et les options contextuelles. La modélisation des processus guidée par l'ontologie devrait pouvoir répondre à cette approche plus orientée vers le contrôle au moyen de classes de "contrôle des processus" dédiées à l'ajout d'un contrôle informatif des processus.

2.3.2 Modèles centrés sur le processus et modèles centrés sur l'objet

Il est possible d'établir une distinction de représentation selon que les données sont modélisées du point de vue du processus ou du point de vue de l'objet, ou des deux. Une perspective processus-et-objet permet de garder le chemin de provenance - l'information sur la façon dont une entité ou son contexte a été modifié dans l'environnement ou dans une chaîne d'approvisionnement. Les processus de transformation qu'il s'agisse d'analyses, de méthodes de traitement des données ou de méthodes de production agricole, de stockage, d'expédition ou de fabrication, relient les états passés, présents et futurs d'une entité, qui peuvent tous être capturés dans un graphe de connaissances. Cela correspond bien aux registres de traçabilité des produits qui documentent la façon dont les produits sont créés ou transformés au cours de leur cycle de vie par divers agents.

Une perspective objet des entités et de leurs caractéristiques à un moment donné peut couvrir le contexte qui / quoi / quand / pourquoi / où d'une situation mais manque d'un cadre pour décrire le "comment".

Une perspective objet est dérivée d'un modèle processus-et-objet en remplaçant tout processus donné par un lien généralisé et donc moins informatif "dérive de" ou "concerne" (entre l'information et une entité) pour relier les entrées et les sorties (i/o). De même, un modèle de processus qui détaille les entités d'entrée et de sortie peut être réduit à un lien de dépendance entre les processus dans une "perspective de processus uniquement" en remplaçant les entités i/o par une relation RO "fournit directement des entrées pour". On perd alors tous les détails concernant les caractéristiques qui auraient pu être associées aux objets auxquels les processus ont été appliqués.

2.3.3 Étapes de processus, parties, dépendances et abstraction

Divers modèles de processus ont une entité "étape", qui, du point de vue du contrôle du processus, est une convention permettant de nommer une vue de processus abstraite et de la classer dans un flux d'opération. Une étape porte la sémantique d'un processus qui peut être isolé comme un événement contrôlable et qui échoue ou réussit, et qui peut avoir des états de pause ou redémarrage. Une étape capture un certain niveau de modularité du processus et de granularité de contrôle. Elle doit être complétée avant de passer à une étape ultérieure qui en est dépendante. Un modèle de processus élégant serait en mesure d'offrir différents niveaux de granularité. Les connexions entre les entités matérielles, les caractéristiques, les processus et les informations à un instant donné sont souvent prises ensemble pour décrire les « états » possibles d'un « système ». Alors que certains modèles n'expriment que des actions à effectuer sur des entités (comme "ébullition"), l'expression des qualités d'entrée et de sortie permet d'exprimer les changements d'états d'un système et les conditions qui doivent être remplies, comme "attendre jusqu'à ce que l'eau soit en ébullition ». Certains modèles permettent aux processus eux-mêmes d'avoir ou d'influencer des caractéristiques qui peuvent être observées ou contrôlées.

Associer une qualité contrôlée ou surveillée à un processus est un raccourci pour modéliser les qualités qui identifient l'entrée ou la sortie du processus - un niveau de détail requis pour l'automatisation industrielle. Les concep-

tions des études expérimentales font une distinction quant à l'entité matérielle ou aux caractéristiques de processus qui sont traitées comme des contrôles (ou paramètres), des variables indépendantes ou des variables dépendantes. Comme condition préalable au succès, le contrôle réel d'une expérience ou les niveaux de variables indépendantes doivent correspondre à la conception ou au plan expérimental, de sorte qu'une étape de contrôle de la qualité implique l'étalonnage précis (mesure) des niveaux de réglage du dispositif de contrôle.

2.3.4 Limites du modèle de processus OWL

L'un des problèmes que pose la logique OWL par rapport au sens commun est que, techniquement, il n'y a pas de moyen facile d'exprimer qu'un processus a directement modifié la qualité d'un matériau d'entrée (ou d'une entité numérique), étant donné qu'il n'est pas possible d'affirmer qu'une instance de matériau de sortie est identique au matériau d'entrée (par exemple, un processus de blanchiment transforme les cheveux foncés d'une personne en cheveux blonds, de sorte que ni les cheveux ni la personne ne peuvent techniquement être considérés comme la "même" entité). Une approche consiste à raisonner sur des observations horodatées concernant des objets, plutôt que sur les propriétés d'un objet directement. On peut aussi créer une instance d'une entité qui est utilisée (manuellement ou par logiciel) pour désigner une entité matérielle donnée n'importe où dans une matrice d'entrée/sortie de processus, marquant effectivement un ensemble d'entités d'instance comme étant à peu près la même entité tout en évitant la logique d'équivalence d'OWL. Cela ne facilitera pas les prouesses en matière de classification avec OWL, mais permettra d'assurer la provenance et la traçabilité en fonction des identifiants attachés aux ressources du processus.

Compte tenu des limites des capacités actuelles des raisonneurs OWL, une ontologie de processus devrait principalement être considérée comme le véhicule permettant de fournir la grammaire des catégories d'entités et des relations qui sont utilisées pour créer des phrases décrivant les processus. Les raisonneurs OWL ne peuvent que déduire l'appartenance à une classe, de sorte que l'ordre des étapes du processus ne peut probablement pas être déduit des dépendances du processus, et des algorithmes extérieurs à la logique OWL sont nécessaires. Certains raisonneurs OWL peuvent traiter des comparaisons simples de propriétés de données numériques (x >=< y), mais l'automatisation dans les laboratoires et les chaînes de montage nécessite de nombreuses opérations de non-classification, telles que l'optimisation des matériaux en lots, le recalibrage des appareils et le calcul des coûts des procédures. Les technologies telles que SWRL [58] ou SPARQL [52] sont adaptées aux calculs de flux d'opérations des graphes de connaissances pilotés par l'ontologie. En créant une séparation entre la validité structurelle et le calcul du flux d'opération, OBO Foundry (et d'autres communautés) peut alors se concentrer sur la mission d'interopérabilité consistant à aligner diverses ontologies de domaine OWL afin d'utiliser un vocabulaire relationnel partagé capable d'exprimer et de valider des structures de données aux niveaux des classes et des instances.

3 Modèle de recette FoodOn

FoodOn intégrera un modèle de processus général lié à l'alimentation afin que les méthodes de transformation des aliments puissent être détaillées dans une série de micro-modèles spécifiques à chaque type d'aliment ou de processus allant de l'acte apparemment simple de cuisiner ou de faire bouillir, à suivre un processus de recette consommateur ou industriel . En raison des limites du raisonnement, l'accent est davantage mis ici sur la recherche de valeur dans des structures de données conformes aux distinctions ontologiques sur le monde, plutôt que sur la réalisation de prouesses de raisonnement sur les données elles-mêmes.

Les patrons de conception de la figure 1 sont une proposition, à discuter, mais une partie est déjà ajoutée à FoodOn.

Actuellement, FoodOn dispose d'un vocabulaire étendu pour décrire la composition d'ingrédients à source unique - la taxonomie et l'anatomie des ingrédients de l'organisme. Pour décrire la composition des aliments à plusieurs composants, FoodOn permet aux classes et instances de produits alimentaires de se référer aux ingrédients (eux-mêmes des produits alimentaires, y compris les additifs) au moyen des relations «has ingredient» et «has defining ingredient».

Ces relations peuvent suffire pour décrire une liste de courses, mais les ingrédients de la recette sont généralement associés à des étapes et doivent apparaître dans l'étiquetage du produit : il faut une liste d'ingrédients, ordonnée en fonction des quantités, données en proportion ou valeur absolue. Souvent, une recette a également besoin de dispositif - même une table ou une cuillère doivent être considérées comme une ressource nécessaire. En même temps, la réutilisation du vocabulaire à différentes échelles de modélisation - de la cuisine au laboratoire en passant par la robotique - tombe, espérons-le, dans des modèles de réutilisation, de sorte que moins de relations sont nécessaires à mesure qu'un modèle de processus générique se développe, l'accent étant davantage mis sur la définition de nouvelles sous-classes de processus, entité, dispositif et information. Le modèle de recette FoodOn proposé dans la la figure 1 comporte un certain nombre de nouveaux composants (représentés par des contours en pointillés), notamment un « ensemble d'ingrédients », un « ensemble d'appareils » et un « ensemble d'instructions » de spécifications d'étape qui permettent respectivement de se référer aux matériaux alimentaires, dispositifs et les processus planifiés qu'ils alimentent. Ces ensembles seront des sous-classes de l'ensemble de données IAO (IAO :0000100) qui est un ensemble de choses du même genre. Une certaine quantité de récursivité du jeu d'instructions est nécessaire lorsque les recettes ont des étapes ou des étapes globales, chaque étape ayant elle-même potentiellement un composant de jeu d'instructions. Chaque spécification d'ingrédient fait référence à une matière alimentaire (produit, produit intermédiaire, ou additif, ou au choix) mais aussi à une mesure proportionnelle, de comptage ou scalaire. Les capacités du dispositif sont indiquées par ses spécifications (taille de la

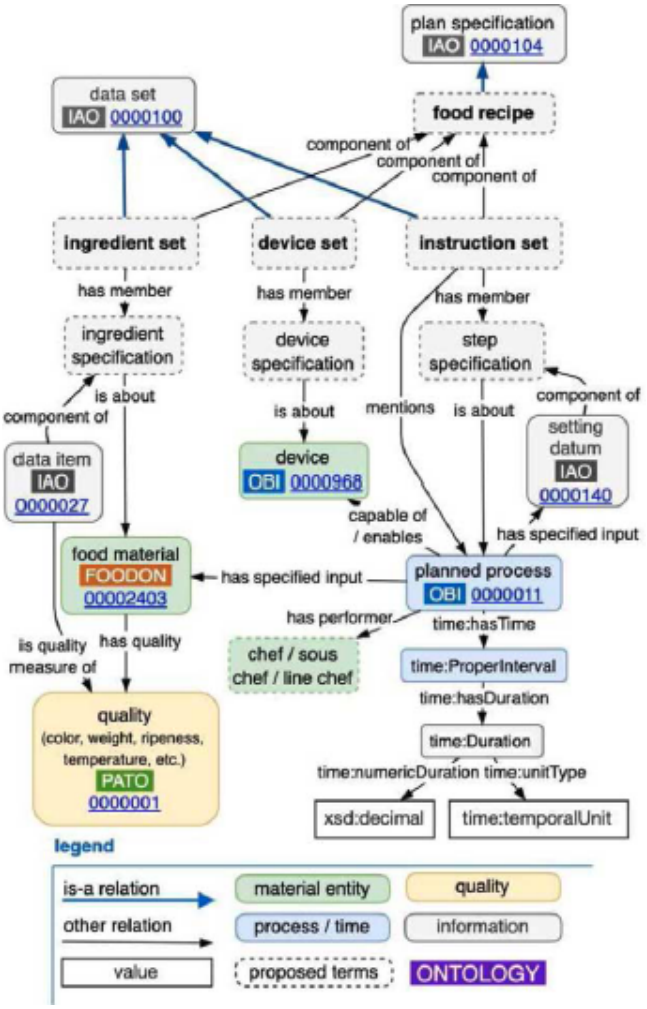


FIGURE 1 – Prototype d'une recette dans FoodOn.

casserole, de la poêle ou du bol, la température nominale du four à vaisselle, etc.). Un commentaire ou une instruction en texte libre peut potentiellement être lié à l'une des entités de la classe en utilisant des annotations telles que "dc :description" ou "rdfs :comment". Le modèle actuel de recette n'inclut pas les procédures de service ou les portions, ni les informations dérivées sur la nutrition ou la durée de cuisson.

4 Besoins ontologiques pour la transformation des aliments

Le cas d'usage, la recette de purée de carottes, décrit amplement dans [11] donne lieu à plusieurs recommandations d'entités et de propriétés à ajouter dans FoodOn ou OBO :

- l'entité **Ingredient set** (Ensemble d'ingrédients) qui est-un data set (ensemble de données) contenant les spécifications des ingrédients ;
- l'entité **device set** (Ensemble de dispositifs) qui est-un data set (ensemble de données) contenant une ou plusieurs spécifications des dispositifs ;
- l'entité **instruction set** (Ensemble d'instructions) qui est-un data set (ensemble de données) contenant les spécifications d'une étape ;
- **Observation** : Une donnée qui est le résultat d'un processus d'observation et qui a des mesures comme composants ;
- l'entité **food recipe** (recette alimentaire) qui est-un plan specifcation (document de spécification) et qui peut avoir des composants qui sont des ensembles d'ingrédients, des ensembles de dispositifs et des ensembles d'instructions ;
- l'entité **ingredient specification** qui spécifie un food material et sa quantité ou son taux ;
- l'entité **device specification** qui spécifie un dispositif et ses réglages ;
- l'entité **step specification** qui spécifie un processus planifié et les restrictions imposées à ses participants.
- la propriété **has ratio** pour représenter une fraction ou un pourcentage ;
- Idéalement, l'élément "has specified output" de OBI devrait être placé sous l'élément "has output" de RO et, de la même manière, l'élément "has specified input" devrait être placé sous l'élément "has input".

Les ustensiles et équipements de cuisine ainsi qu'une longue liste de processus alimentaires seront également nécessaires, mais ils pourront être ajoutés progressivement à FoodOn ou à l'ontologie EO (Environnement Ontology) [**?**] (qui contient les produits manufacturés) ou à une autre ontologie, selon les besoins. Ce modèle peut être étendu dans un certain nombre de directions - en le reliant à des informations sur la nutrition, le risque d'allergie et les substitutions d'ingrédients et l'évaluation de l'empreinte écologique. Les pratiques et représentations culturelles pourraient être incluses et les réglages des appareils pourraient être spécifiés. Une piste de recherche consiste à déterminer la logique appropriée ou les informations nécessaires pour déduire les étapes à partir d'un graphe de dépendance d'un modèle de recette. De même, l'abstraction des recettes pourrait être possible grâce à une transformation qui conduirait aux processus de niveau supérieur et à tous les ingrédients sous-jacents. La validation du modèle pourrait être réalisée en le testant sur les projets qui ont des bases de données de recettes.

5 Conclusion

La modélisation des processus permet d'obtenir le liant nécessaire pour expliquer l'histoire des choses matérielles et des informations dérivées. La transformation des aliments - depuis le point de récolte agricole et au-delà - est un sujet de préoccupation immédiate à l'heure de l'adaptation au climat, de la mondialisation de la transformation et de la distribution des aliments, des déserts alimentaires et des risques sanitaires liés à l'alimentation - obésité, malnutrition, contaminations et agents pathogènes d'origine alimentaire. Une ontologie de processus générique cohérente, avec des composants spécialisés, permettra plus d'interopérabilité entre la recherche en science alimentaire et les systèmes

opérant au sein de la chaîne agro-alimentaire, pour une transition vers un modèle plus sain et plus durable. Notre analyse des lacunes de la capacité de modélisation des processus basé sur le modèle OBO actuel, fournit une feuille de route pour l'ajout de nouveaux termes essentiels pour la construction de micromodèles permettant de représenter des recettes, ainsi que pour la recherche et l'automatisation dans le domaine de l'alimentation. Les principaux axes de recherche restent l'étude des qualités organoleptiques et physiques des aliments (kinesthésie, rhéologie et gastronomie moléculaire, par exemple), afin de pouvoir décrire, classer et prédire leur transformation à la suite de processus naturels et planifiés. Nous espérons également que notre travail sur les modèles de processus OBO sera pertinent au-delà du domaine alimentaire. Les commentaires sur les exigences en matière de modélisation de la transformation des aliments, sur la sémantique plus large des modèles de processus et la participation à cet effort sont les bienvenus.

Références

[1] Github.

[2] Cross-domain integration.

[3] Obo foundry principles : Overview.

[4] Submissions :food recipe ingredient substitution ontology design pattern - odp.

[5] Using the process pattern ? ddi 4.0 dev documentation.

[6] Welcome to the data documentation initiative.

[7] E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Cech, J. Chilton, D. Clements, N. Coraor, B. A. Grüning, A. Guerler, J. Hillman-Jackson, S. Hiltemann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko, and D. Blankenberg. The galaxy platform for accessible, reproducible and collaborative biomedical analyses : 2018 update. *Nucleic acids research*, 46(W1) :W537 ?W544, Jul 2018.

[8] R. Arp, B. Smith, and A. D. Spear. *Building Ontologies with Basic Formal Ontology*. MIT Press, Jul 2015.

[9] A. Bandrowski, R. Brinkman, M. Brochhausen, M. H. Brush, B. Bug, M. C. Chibucos, K. Clancy, M. Courtot, D. Derom, M. Dumontier, L. Fan, J. Fostel, G. Fragoso, F. Gibson, A. Gonzalez-Beltran, M. A. Haendel, Y. He, M. Heiskanen, T. Hernandez-Boussard, M. Jensen, Y. Lin, A. L. Lister, P. Lord, J. Malone, E. Manduchi, M. McGee, N. Morrison, J. A. Overton, H. Parkinson, B. Peters, P. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, R. H. Scheuermann, D. Schober, B. Smith, L. N. Soldatova, J. Stoeckert, Christian J., C. F. Taylor, C. Torniai, J. A. Turner, R. Vita, P. L. Whetzel, and J. Zheng. The ontology for biomedical investigations. *PloS one*, 11(4) :e0154556, Apr 2016.

[10] S. Dervaux, J. Dibie, L. Ibanescu, and J. Raad. Po2 process and observation ontology, 2021.

[11] D. Dooley, M. Weber, L. Ibanescu, M. Lange, L. Chan, L. Soldatova, C. Yang, R. Warren, C. Shimizu, H. K. McGinty, and W. Hsiao. Food process ontology requirements. *Semantic Web*, Preprint(Preprint) :1–32, 2022. Publisher : IOS Press.

[12] D. M. Dooley, E. J. Griffiths, G. S. Gosal, P. L. Buttigieg, R. Hoehndorf, M. C. Lange, L. M. Schriml, F. S. L. Brinkman, and W. W. L. Hsiao. Foodon : a harmonized food ontology to increase global food traceability, quality control and data integration. *NPJ science of food*, 2 :23, Dec 2018.

[13] L. Elizabeth, P. Machado, M. Zinöcker, P. Baker, and M. Lawrence. Ultra-processed foods and health outcomes : A narrative review. *Nutrients*, 12(7), 2020.

[14] Y. He, Z. Xiang, J. Zheng, Y. Lin, J. A. Overton, and E. Ong. The extensible ontology development (xod) principles and tool implementation to support ontology interoperability. *Journal of biomedical semantics*, 9(1) :3, Jan 2018.

[15] R. C. Jackson, N. Matentzoglu, J. A. Overton, R. Vita, J. P. Balhoff, P. L. Buttigieg, S. Carbon, M. Courtot, A. D. Diehl, D. Dooley, W. Duncan, N. L. Harris, M. A. Haendel, S. E. Lewis, D. A. Natale, D. Osumi-Sutherland, A. Ruttenberg, L. M. Schriml, B. Smith, C. J. Stoeckert, N. A. Vasilevsky, R. L. Walls, J. Zheng, C. J. Mungall, and B. Peters. Obo foundry in 2021 : Operationalizing open data principles to evaluate ontologies. Jun 2021.

[16] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois. Sosa : A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, 56 :1–10, 2019.

[17] D. Knorr and M. Augustin. Food processing needs, advantages and misconceptions. *Trends in Food Science Technology*, 108 :103–110, 2021.

[18] S. Little and C. Cox. Extensions to the owl-time ontology - entity relations, Jul 2020.

[19] N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin. The iplant collaborative : Cyberinfrastructure for enabling data to discovery for the life sciences. *PLoS biology*, 14(1) :e1002342, Jan 2016.

[20] P. Missier, K. Belhajjame, and J. Cheney. The w3c prov family of specifications for modelling provenance metadata. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT 13, pages 773–776, New York, NY, USA, Mar 2013. Association for Computing Machinery.

[21] OriginTrail. Gs1 digital link : A gateway towards trillions of digital twins ?, Aug 2020.

[22] O. X. Service. Core ontology for biology and biomedicine < ontology lookup service < embl-ebi.

[23] L. N. Soldatova, D. Nadis, R. D. King, P. S. Basu, E. Haddi, V. Baumlé, N. J. Saunders, W. Marwan, and B. B. Rudkin. Exact2 : the semantics of biomedical

protocols. *BMC bioinformatics*, 15 Suppl 14 :S5, Nov 2014.

[24] A. Ławrynowicz, A. Wróblewska, W. T. Adrian, B. Kulczyński, and A. Gramza-Michałowska. Food recipe ingredient substitution ontology design pattern. *Sensors (Basel, Switzerland)*, 22(3) :1095, Jan 2022.

Session 3 : Alignement d'ontologies et liage de données

Amélioration de l'alignement de propriétés d'ontologies grâce aux plongements et à l'extension d'alignement

Guilherme Sousa[1], Rinaldo Lima[2], Cassia Trojahn[1]

[1] Institut de Recherche en Informatique de Toulouse, Toulouse, France
[2] Universidade Rural de Pernambuco, Recife, Brazil

guilherme.santos-sousa@irit.fr, cassia.trojahn@irit.fr, rinaldo.jose@ufrpe.br

Résumé

Les approches d'alignement de propriétés de schémas de graphes de connaissances restent en retrait par rapport à la mise en correspondance des classes. Les propriétés impliquent souvent une variation plus importante dans leur dénomination (variation du verbe, mots fonctionnels, synonymes) que les classes. Cet article propose une approche d'alignement de propriétés qui combine les plongements et les extensions d'alignement afin d'améliorer les performances de la mise en correspondance de ce type d'entité. L'approche proposée est compétitive par rapport aux systèmes d'alignement existants.

Mots-clés

Alignement d'ontologies, alignement de propriétés, apprentissage automatique, plongements de mots

Abstract

Approaches for matching properties in knowledge graph schemas still behavior behind the matching of classes. Properties frequently involve a higher variation in naming (verb variation, functional words, common synonyms) than classes. This paper proposes a property-matching approach that combines embeddigns and alignment extension to improve the property matching performance. The proposed approach performs competitively with state-of-the-art alignment systems on well-known benchmarks in the field.

Keywords

Ontology matching, property alignment, machine learning, embeddings

1 Introduction

L'objectif du processus d'alignement d'ontologies est de trouver des correspondances entre les entités de différentes ontologies, généralement deux ontologies. L'une des principales tâches de ce processus consiste à trouver des correspondances entre propriétés. Une métrique courante pour trouver des correspondances entre propriétés est la métrique de similarité de chaînes de caractères, par exemple, la distance d'édition, comparant les étiquettes des entités. Cependant, l'utilisation de telles métriques ne permet pas de rappeler une partie des correspondances et ne permet pas de filtrer les entités homonymes puisqu'elles partagent les mêmes étiquettes mais ont des significations différentes [1, 22]. Récemment, les modèles de plongements ont attiré l'attention dans le domaine de l'alignement des ontologies et ont été appliqués dans plusieurs systèmes tels que TOM [15], Fine-Tom [14], ALOD2Vec [20], et AMD [27]. Toutefois, ces techniques se sont révélées utiles lorsqu'elles sont combinées à d'autres stratégies de mise en correspondance.

Les plongements de mots statiques couramment utilisés dans l'alignement d'ontologies, comme Glove [19] et Word2Vec [18], ont cependant des problèmes pour modéliser la notion de similarité. Cela est souligné dans les travaux sur l'analyse des sentiments [30, 13], car ils accordent une similarité forte à des mots présents dans le même contexte, tels que "Day" et "Night" qui ont, en fait, des significations opposées. Ce problème est accentué lors de l'utilisation des plongements statiques pour représenter les phrases, car le sens des mots change en fonction de leur contexte. Pour aider à résoudre ce problème, les modèles contextuels comme BERT [9] sont capables de générer différents plongements pour le même mot en fonction du contexte de la phrase, ce qui permet d'obtenir de meilleurs plongements de phrases. En particulier, pour la tâche d'alignement d'ontologies, ces modèles sont utiles pour représenter les informations textuelles dans les ontologies, puisque les ontologies peuvent être considérées comme un graphe de concepts dont les caractéristiques des nœuds sont représentées dans le texte en langue naturelle (dans les étiquettes et les annotations). En outre, ces modèles facilitent l'application des techniques de plongement de graphes, car ils nécessitent des vecteurs de caractéristiques de longueur fixe pour le traitement, alors que les caractéristiques textuelles des ontologies peuvent être de longueur variable, comme le montre la figure 1. Un exemple d'approche d'alignement qui illustre cette stratégie est DAEOM [29], où BERT [9] est utilisé pour extraire des vecteurs de caractéristiques de taille fixe à partir du contenu textuel de l'entité. Ces caractéristiques sont utilisées dans un réseau de neurones de graphes [24] pour mieux contextualiser les caractéristiques des nœuds. Cependant, comme cette approche est supervisée et qu'aucun jeu de données d'alignement de grande taille n'est disponible pour affiner ces modèles, ces approches ont des difficultés à atteindre des meilleures performances.

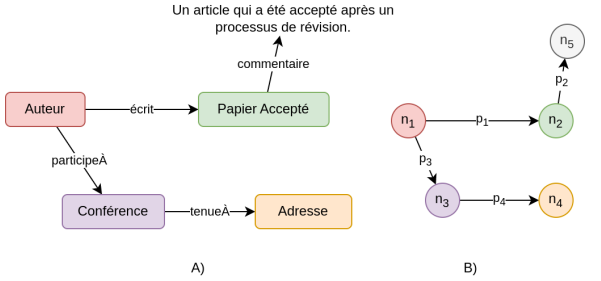


FIGURE 1 – A) Les entités de l'ontologie sont nommées à l'aide d'un texte en langage naturel de longueur variable. B) Les réseaux de neurones de graphe nécessitent des vecteurs de caractéristiques de longueur fixe qui doivent être générés pour pouvoir travailler sur des ontologies.

Cet article aborde le problème de l'alignement de propriétés à l'aide de plongements pré-entraînés et de l'extension de l'alignement [12], en étendant le système PropString [5]. Comme PropString n'utilise que des métriques de similarité lexicale, l'ajout des plongements pré-entraînés donne au système plus de flexibilité pour représenter des entités similaires avec des étiquettes différentes, tout en réduisant le besoin d'un réglage fin lorsqu'ils sont appliqués dans des étapes spécifiques. En complément, l'extension de l'alignement s'est avérée utile pour capturer les patrons d'alignement fréquents lorsque des propriétés similaires sont détectées. Le système proposé est évalué sur des jeux de données utilisés dans le cadre de la campagne OAEI. Les résultats montrent que l'utilisation des techniques proposées peut améliorer l'alignement des propriétés par rapport aux meilleurs systèmes participant à la track Conférence d'OAEI.

Le reste de l'article est organisé comme suit. La section 2 présente la définition du problème et la définition des représentations des propriétés considérées dans ce travail. La section 3 détaille l'architecture du système ainsi que les techniques utilisées. La section 4 présente les expériences menées pour évaluer la performance du système. La section 5 discute les travaux liés et, enfin, la section 6 conclut l'article.

2 Définition du problème

L'alignement de propriétés consiste à chercher de propriétés similaires entre deux ontologies différentes. Cette tâche peut être définie comme la recherche du meilleur ensemble de correspondances de propriétés A étant donné les ontologies en entrée O_1 et O_2. Dans cet article, nous définissons les propriétés comme toutes les entités S qui satisfont le prédicat $P(S) : \exists!D, \exists!R, domain(S, D) \wedge range(S, R)$. Compte tenu de cette définition, la tâche d'alignement est définie comme suit : étant donné la fonction de similarité des propriétés Sim, trouver l'ensemble de correspondances $A = \{(p_1, p_2) \in O_1 \times O_2 | Sim(p_1, p_2) > t\}$ produit en mesurant la similarité de chaque combinaison de paires de propriétés dans l'ontologie source et l'ontologie cible et en sélectionnant celles dont la similarité est supérieure à un seuil t donné. La Figure 2 présente l'architecture générale pour le calcul de similarité, mise en œuvre par la plupart de systèmes.

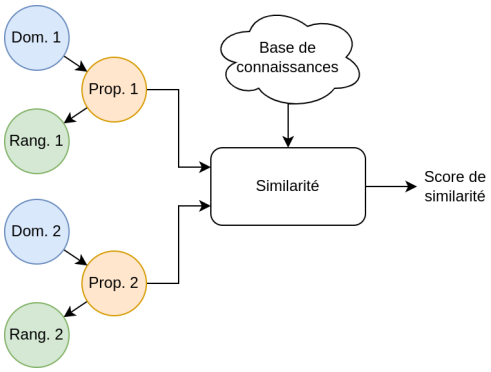


FIGURE 2 – Architecture générale pour le calcul de similarité. Étant donné deux propriétés avec leurs domaines (*rdfs :domain*) et leurs portées (*rdfs :range*), la fonction de similarité génère un score qui mesure leur degré de similarité.

L'une des difficultés rencontrées par les systèmes d'alignement de propriétés pour s'adapter à des domaines distincts réside dans les différentes représentations des propriétés. Par exemple, les ontologies peuvent représenter les propriétés en tant qu'entité d'un type tel que *owl :ObjectProperty* lié à une entité de domaine par le prédicat *rdfs :domain* et à une entité de la portée par le prédicat *rdfs :range*. Dans les graphes de connaissances, les propriétés sont plutôt représentées comme des prédicats qui relient un sujet à un objet. Dans ce cas, la même propriété peut relier des entités ayant une sémantique différente, même dans la même ontologie. Par exemple, *Author* et *Color* peuvent avoir une propriété *name* dans le même graphe de connaissances. Cet exemple montre que, dans ce cas, le domaine de cette propriété est composé de plusieurs entités, ce qui accroît la complexité de la fonction de similarité, car elle doit tenir compte de la manière de mesurer la similarité du domaine composé d'un groupe d'entités.

3 Approche proposée

L'approche proposée dans cet article est basée sur le système PropString intégrant les plongements et l'extension d'alignement. L'hypothèse générale du système original PropString concernant les propriétés similaires est qu'elles doivent avoir des domaines, des portées et des étiquettes similaires. La métrique utilisée pour mesurer la similarité entre les domaines et les portées est la métrique TF-IDF (*Term Frequency-Inverse Document Frequency*), qui s'est avérée être la meilleure métrique lexicale pour l'alignement des classes (dans leur proposition). Le TF-IDF est une métrique couramment utilisée pour la recherche d'information et repose sur l'hypothèse que les mots rares partagés entre deux objets les rendent similaires et que les termes fréquents qui apparaissent à de nombreux endroits ne sont

pas importants. Cette métrique suppose une hypothèse statistique sur la similarité, et puisqu'il s'agit d'une métrique globale, elle compte la fréquence des mots dans toutes les entités de l'ontologie.

Afin d'aligner les étiquettes de propriété, une version souple de TF-IDF est appliquée avec JaroWinkler comme métrique de similarité pour inclure également les mots lexicalement similaires dans les vecteurs de fréquence. Cette métrique est appliquée au concept central de l'étiquette de propriété introduit dans PropString [5]. Ce concept a été conçu par les auteurs de PropString en analysant les modèles de dénomination de propriété communs. Le concept central se compose du premier verbe de plus de quatre caractères ou, si ce verbe n'est pas trouvé, du premier nom et des adjectifs qui les accompagnent. Pour trouver le concept central, un étiqueteur POS est utilisé. Si la valeur minimale entre la similarité du domaine, de la portée et du concept central dépasse un certain seuil, les propriétés sont considérées comme similaires.

L'utilisation des similarités de domaine et de portées aide le système à filtrer les alignements de propriété faussement positifs. Un exemple de faux alignement possible est la propriété "writes" avec le domaine "Author" et la portée "Paper" dans l'ontologie O_1 et la propriété "writtenBy" avec le domaine "Paper" et la portée "Author" dans l'ontologie O_2. Les deux propriétés "writes" et "writtenBy" ont le même préfixe et peuvent avoir une grande similitude avec certaines métriques lexicales, mais la prise en compte des domaines et des portées donne une faible valeur de similarité à cette paire de propriétés en raison de leur différence.

Les améliorations proposées à PropString reposent sur deux observations principales. La première observation est que dans certains cas, le système PropString trouve des étiquettes et des portées de propriétés avec une grande similarité, mais la similarité de domaine est nulle. Ce cas se produit parce que les deux domaines sont synonymes de mots différents et que la métrique lexicale leur donne une faible similarité. Dans ce cas, des approches plus robustes sont nécessaires pour récupérer ces correspondances. Une deuxième observation est que les propriétés inverses des propriétés alignées sont plus susceptibles d'avoir une correspondance entre elles [12]. L'extension de l'alignement repose sur le principe de localité qui stipule que les entités proches d'entités précédemment mises en correspondance sont susceptibles d'être similaires, et les correspondances établies peuvent donc être utilisées pour détecter les correspondances potentielles entre les entités proches dans le voisinage du graphe. En ce sens, nous pouvons étendre les correspondances en incluant un alignement entre les inverses des propriétés comparées s'ils existent et si les propriétés comparées sont similaires. Cette approche peut améliorer le rappel du système car elle récupère les correspondances avec des relations sémantiques complexes qui seront incluses si elles ont des inverses 'alignables' plus simples.

Dans les sections suivantes, nous présentons l'architecture du système proposé et les modifications que nous avons apportées à PropString, y compris l'utilisation des plongements et des extensions de l'alignement. L'architecture est

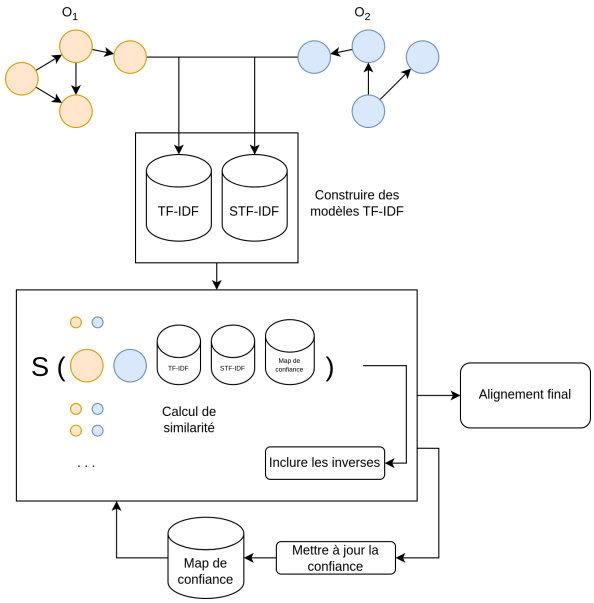


FIGURE 3 – Architecture du système proposé.

présentée dans la Figure 3 et se compose de deux éléments principaux : la construction du modèle TF-IDF et le calcul de la similarité.

3.1 Construction de modèles TF-IDF

Étant donné que les métriques TF-IDF et Soft TF-IF (décrites ci-dessous) sont globales, elles prennent en compte la fréquence des mots dans toutes les informations relatives aux étiquettes des entités dans les ontologies source et cible afin de construire les vecteurs de fréquence. De ce fait, la construction des modèles intervient avant le processus de mise en correspondance.

La métrique de similarité du modèle est composée du TF-IDF et du Soft TF-IDF. Le modèle TF-IDF est utilisé pour calculer la similarité entre les étiquettes de domaine et les étiquettes des portées. Pour construire le TF-IDF et sa version souple, un document virtuel est créé pour chaque entité dans les deux ontologies, contenant des informations relatives à chaque entité, puis les vecteurs sont calculés. Le document virtuel généré pour les classes de l'ontologie est composé du nom de la classe et, pour les propriétés, du nom de la propriété, du domaine et des portées. En cas de domaines et de portées multiples, toutes les valeurs sont ajoutées au document virtuel. Dans tous les cas, les étiquettes sont divisées à l'aide d'un tokenizer capable de diviser les conventions de nommage en camel case [2] ou en underscore [2]. A la fin de ce processus, les étiquettes sont mises en minuscules et jointes pour constituer le document final. Par exemple, la propriété *writePaper* avec le domaine *Author* et la portée *Paper* produit le document "author write paper paper paper". L'ensemble des documents est utilisé pour construire les modèles de fréquence, le vocabulaire et l'IDF. La IDF utilisée pour la métrique générale est énoncée dans l'équation 1, où N est le nombre de documents et $df(t)$ est le nombre de documents dans lesquels le terme t

apparaît.

$$idf(t) = \log(\frac{N+1}{df(t)+1}) + 1 \quad (1)$$

Après avoir construit le vocabulaire et calculé les valeurs IDF, la similarité entre les documents est mesurée en calculant le vecteur de fréquence des termes et en multipliant chaque mot par son IDF respectif. Les vecteurs sont ensuite normalisés et la similarité finale est calculée à l'aide de la similarité cosinus, soit d_1 et d_2 deux vecteurs comme décrit dans l'équation 2.

$$sim(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1|| \cdot ||d_2||} \quad (2)$$

Pour les étiquettes de propriétés, le système utilise le Soft TF-IDF. Le Soft TF-IDF inclut également les mots similaires qui dépassent un seuil de score de similarité dans le décompte des fréquences. Le Soft TF-IDF est construit en utilisant la métrique de Jaro-Winkler avec un seuil de 0.8.
Il est à noter que le système ne fonctionne qu'avec l'alignement monolingue des propriétés en anglais.

3.2 Plongements dans la similarité

Après la génération des modèles TF-IDF, le système calcule le score de similarité pour chaque paire de propriétés. Le score de similarité final est le minimum de trois valeurs de similarité basées sur les similitudes de domaine, de la portée et d'étiquette de propriété. Tout d'abord, la similarité de domaine est calculée à l'aide des vecteurs TF-IDF, et une similarité de plongement est utilisée comme solution de repli lorsque la métrique donne une similarité nulle. Cette approche est plus fiable lorsque les domaines comparés ne contiennent qu'un seul mot. Ainsi, dans ce cas, la similarité cosinus entre les plongements des mots du domaine remplace la similarité du domaine en utilisant les plongements pré-entraînés de la Finnish Internet Parsebank [17]. La similarité de la portée est calculée selon les mêmes étapes que le calcul de similarité de domaine, sans le repli de plongement, en filtrant d'abord les adjectifs de l'étiquette de la portée.
La première étape du calcul de la similarité des étiquettes de propriété consiste à retirer le dernier mot de l'étiquette de propriété s'il est égal au premier mot de l'étiquette de la portée. Par exemple, la propriété *hasPaper* avec le domaine *Author* et la portée *Paper* produira la phrase "Author has Paper Paper". Après le traitement, la phrase "Author has Paper" est retenue. Ensuite, l'étiquette de propriété est étiquetée à l'aide d'un marqueur POS et le système utilise le concept central pour calculer la similarité à l'aide des vecteurs Soft TF-IDF. Le premier verbe avec plus de quatre caractères (cette heuristique est appliquée pour filtrer les verbes anglais courts comme "has" ou "let" et est considérée après l'analyse des patrons de noms de propriétés [5]) ou le nom avec ses adjectifs, si aucun verbe n'est trouvé, est utilisé comme entité centrale de l'étiquette de propriété. Le système applique la même stratégie que celle décrite précédemment pour la similarité des domaines aux étiquettes de propriété. Dans ce cas, un modèle de similarité des phrases

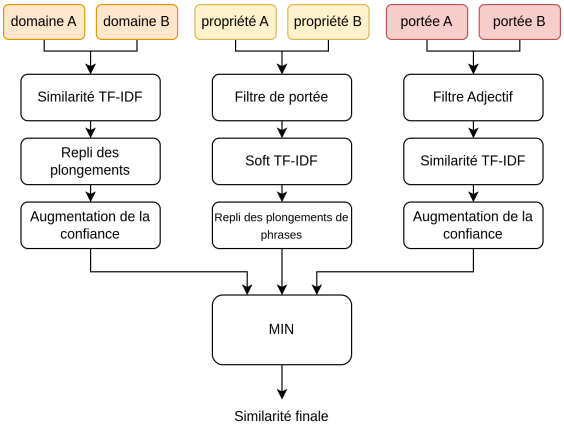


FIGURE 4 – Pipeline de similarité des propriétés.

est utilisé pour générer des représentations de plongement des étiquettes de propriété, et ces représentations sont comparées à l'aide de la similarité cosinusoïdale. Le repli se produit lorsque la similarité de domaine et de la portée est supérieure à 0.9 et que la similarité d'étiquette est inférieure à 0.1. Le modèle de similarité de phrases [25] utilisé dans ce travail provient du référentiel HuggingFace [28]. La phrase utilisée est composée de l'étiquette de propriété concaténée avec des étiquettes de la portée. Dans le cas des propriétés d'objet, l'étiquette de la classe présente dans la portée est utilisée. Une illustration de la procédure de calcul de la similarité est présentée dans la Figure 4.
En raison des différentes structures syntaxiques présentes dans les étiquettes de propriétés et les étiquettes de classe, deux modèles de plongement distincts ont été appliqués. La majorité des propriétés contiennent des verbes qui sont mieux capturés à l'aide de modèles de plongement de phrases en raison de la distribution similaire qu'ils ont avec les données d'apprentissage. Les tests empiriques appliquant les plongements de phrases aux étiquettes de classe donnent de moins bons résultats. Pour cette raison, l'incorporation de mots est appliquée lorsque la similarité du domaine est calculée puisque les domaines de propriété sont des classes.

3.3 Extension de l'alignement

Après le calcul final de la similarité des propriétés, le système ajoute la paire de propriétés à l'ensemble de correspondances si la valeur de similarité dépasse le seuil donné. Si certaines des correspondances précédentes contiennent l'une des propriétés alignées, seule la paire de domaines présentant une similarité élevée est conservée. En outre, sur la base du principe de localité, les inverses des propriétés alignées sont inclus dans l'ensemble d'alignement final car ils sont plus susceptibles d'être alignés. Basé également sur le principe de localité, les alignements sont encore étendus en utilisant les informations structurelles des propriétés concernant les domaines. Cette extension est réalisée en maintenant une carte de confiance qui stocke la similarité entre les paires de domaines qui augmentent la similarité

des propriétés entre les domaines de propriété précédemment alignés. La carte de confiance est une structure clé-valeur dans laquelle la clé est une paire d'entités et la valeur est la similarité entre elles. Il est utilisé dans l'étape de calcul de similarité pour augmenter la similarité des domaines s'ils apparaissent comme des domaines dans des propriétés précédemment alignées. Ensuite, le système augmente la similarité de la paire de domaines de propriétés alignées dans la carte de confiance de 0.66 (établi empiriquement). Comme la carte de confiance est mise à jour, plusieurs étapes sont nécessaires pour découvrir de nouvelles correspondances basées sur ce processus d'inférence, car les calculs de similarité incluront la valeur de similarité mise à jour.

4 Expérimentations

4.1 Évaluation sur le jeu de données *Conférence*

La première série d'expériences a été menée sur le jeu de données Conférence [1] disponible dans le cadre des campagnes d'évaluation OAEI. Ce jeu de données consiste en 21 paires d'alignements entre 7 ontologies. Sur les 21 paires, 7 alignements de référence ne contiennent aucune propriété. L'évaluation ne prend en compte que les paires qui contiennent des alignements de référence de propriété. La performance du système est évaluée pour chaque paire individuellement et en considérant le résultat global qui est le total des alignements dans toutes les paires d'alignements. Les résultats globaux des systèmes évalués peuvent être consultés sur la page OAEI 2021 [2]. Les résultats pour chaque paire sont calculés à partir des alignements de référence basés sur les alignements produits par les systèmes participants pour l'année 2021 et sont équivalents à l'évaluation ra1-M2 (alignements de référence entre propriétés). Les systèmes de référence sont ceux qui ont participé à la campagne et sont comparés à l'implémentation de base équivalente à PropString et à la version améliorée (appelé ici PropMatch) disponible sur le Gitlab de l'IRIT sur licence MIT [3]. Le seuil de similarité utilisé par le système proposé dans toutes les évaluations est de 0.65. Le système obtient la meilleure métrique F-measure avec cette valeur lorsqu'il a été testé avec un seuil allant de 0 à 1 par pas de 0.05 avec toutes les modifications. La progression des performances est illustrée dans la Figure 5.

PropMatch a été implémenté en Python à partir de l'implémentation Java originale de PropString et amélioré par l'ajout des modifications décrites dans la section 3.2. Pour cette expérience, l'impact progressif de chaque modification est évalué dans les mêmes conditions, ainsi que le nombre de comparaisons totales dans toutes les paires d'ontologies. Le résultat de l'évaluation est présenté dans la Table 1.

Le système proposé a également été comparé aux systèmes qui ont participé à OAEI en 2021, ainsi que avec PropString. Les résultats sont présentés dans la Table 2. PropMatch surpasse le meilleur système AML de 11% en rappel et 6% en F-mesure. Il a une moins bonne précision par rapport à PropString et à AML. Ce résultat est dû à l'impact de l'utilisation de plongements et de l'inclusion d'inverses, comme cela a été observé dans les résultats des modifications progressives. Les plongements utilisés ont donné des valeurs de similarité élevées à des éléments qui pouvaient être liés mais qui n'étaient pas similaires, ce qui a entraîné une baisse de la précision du système. Un autre problème est l'inclusion aveugle des inverses, car certains inverses ne sont pas présents dans l'alignement de référence, et ce schéma peut ne pas être présent dans toutes les formulations de l'ontologie.

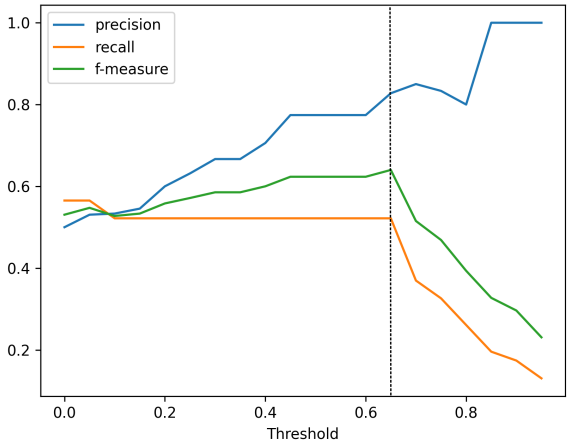


FIGURE 5 – Progression de la précision, du rappel et de la F-mesure à mesure que le seuil augmente.

Enfin, le système proposé a été comparé, pour chaque paire d'ontologies, aux systèmes AML et LogMap, qui ont obtenu la meilleure F-mesure dans l'alignement des propriétés. Comme le montre la Table 3, PropMatch obtient la meilleure F-mesure dans 8 paires, AML dans 6 paires et LogMap dans 4 paires d'alignement. Il peut récupérer les alignements dans 12 paires, tandis que AML et LogMap récupèrent les alignements dans 6 paires. AML atteint une précision de 100% dans toutes les paires d'alignements où un alignement est trouvé, tandis que LogMap et le système proposé peuvent atteindre une précision de 100% dans 67% des paires d'alignements. Aucun des systèmes ne peut récupérer d'alignements entre Conference et Iasted. Dans cette paire d'alignements, le seul alignement de propriétés existant est entre *Person contributes Conference_document* et *Person write Item*, décrits respectivement dans la propriété de domaine et de la portée. La similarité des chaînes lexicales ne permet pas de retrouver la similarité entre les libellés de *contributes* et *write* car leur similarité lexicale est faible, et il en va de même pour *Conference_document* et *Item*. En fait, dans l'ontologie Iasted, la classe *Item* a une sous-classe nommé *Document* qui peut être utilisée comme information pour récupérer cette correspondance.

1. http ://oaei.ontologymatching.org/2021/conference/index.html
2. http ://oaei.ontologymatching.org/2021/results/conference/index.html
3. https ://gitlab.irit.fr/melodi/ontology-matching/propmatch

Description	Précision	Rappel	F-mesure
PropString	**1.00**	0.28	0.44
Utilisation des plongements dans la similarité des domaines	0.84	0.35	0.49
Addition d'inverses	0.81	0.39	0.53
Similitude de plongements des phrases appliquée aux étiquettes des propriétés	0.68	0.41	0.51
Filtre des mots répétés et les adjectifs dans les étiquettes des portées	0.71	0.48	0.57
similarité des domaines pour les propriétés précédemment alignées	0.73	**0.52**	0.61
Limitation de la cardinalité (1-1)	0.83	**0.52**	**0.64**

TABLE 1 – Progression des performances avec l'application de modifications.

Nom	Précision	Rappel	F-mesure
PropMatch	0.83	**0.52**	**0.64**
AML	**1.0**	0.41	0.58
PropString	**1.0**	0.28	0.44
LogMap	0.62	0.28	0.39
GMap	0.56	0.2	0.29
Wikitionary	0.24	0.28	0.26
TOM	0.27	0.24	0.25
ALOD2Vec	0.22	0.3	0.25
LogMapLt	0.24	0.22	0.23
FineTOM	0.24	0.22	0.23
OTMapOnto	0.13	0.48	0.2
edna	0.21	0.11	0.14
StringEquiv	0.07	0.02	0.03

TABLE 2 – Résultats obtenus par les systèmes participant à l'OAEI 2021.

4.2 Évaluation sur d'autres jeux de données

L'autre jeu de données de teste est celui des *OAEI knowledge graph*[4]. Dans ce jeu de données, huit graphes de connaissances sont alignés en 5 paires. Ces graphes de connaissances représentent les propriétés comme un prédicat qui relie une instance à une valeur de la portée. Mais dans cette structure, plusieurs instances peuvent partager la même propriété. Par exemple, des auteurs, des films et des entreprises peuvent avoir le même nom de propriété avec des portées différentes. La propriété peut être considérée comme ayant un domaine et une portée complexes. Le système sélectionne la paire domaine/portée la plus fréquente contenant les types d'instances pour servir de domaine et de portée uniques pour la propriété. L'étape de chargement du graphe effectue ces transformations afin que le système puisse voir la même structure pour différentes représentations des propriétés.

Le système est comparé à ceux qui ont participé à la campagne OAEI pour la track graphe de connaissances, en fonctionnant avec une valeur de seuil de 0.0 et avec 1 itération. Les résultats sont présentés dans la Table 5. Le seuil sélectionné était 0 car c'est le seuil qui produit des correspondances. Comme on peut le voir dans les résultats, sur les 8 systèmes évalués, seuls 4 systèmes ont été capables de produire des alignements, le système proposé étant l'un d'entre eux. En ce sens, PropMatch obtient de meilleurs résultats globaux entre les jeux de données qu'AMD et LogMap. Cependant, il est possible d'observer qu'il ne peut pas atteindre la performance de base dans cette track. Ce fait est dû à l'utilisation par le système de la valeur de similarité minimale entre le domaine, la portée et la similarité des étiquettes, ce qui rend difficile l'alignement du domaine et de la portée, même lorsque les étiquettes des propriétés sont identiques.

Afin de mieux analyser l'impact de l'utilisation des similitudes, nous avons testé PropMatch avec différentes combinaisons de similitudes, toutes avec un seuil de 0. Quatre combinaisons de similarité ont été testées : uniquement les étiquettes de propriété (p), le domaine et la propriété (d+p), la propriété et la portée (p+r) et la configuration de base qui prend en compte toutes les combinaisons (d+p+r). Les résultats de cette évaluation sont présentés dans la Table 4. Il est possible de voir dans les résultats que l'utilisation de la seule similarité d'étiquette de propriété (p) permet d'obtenir des résultats similaires en termes de rappel aux systèmes les plus performants, avec une précision réduite en raison de la faible valeur du seuil. Cependant, avec l'ajout des similitudes de domaine et de la portée (d+p, p+r, et d+p+r), les performances du système diminuent.

Comme nous l'avons vu ci-dessus, une étape de pré-traitement doit être appliquée pour que le système puisse traiter des domaines et des portées complexes. En raison de cette étape de pré-traitement, le système reçoit une seule paire domaine/portée qui peut ne pas représenter toutes les combinaisons possibles décrites par la propriété. De plus, étant donné que l'alignement de chaînes obtient des résultats élevés en ne tenant compte que des étiquettes des propriétés, le repli des plongements se produit rarement, de sorte que l'utilisation des plongements par le système n'a qu'un faible impact dans ce domaine. Un autre problème est que, aucune propriété inverse n'est présente, de sorte que la stratégie d'extension de l'alignement n'est pas appliquée. Les domaines sont complexes et la stratégie proposée ne peut pas représenter les informations nécessaires pour produire une comparaison de similarité suffisante.

5 Travaux liés

L'un des premiers travaux comparant les performances de différentes techniques basées sur les chaînes de caractères dans l'alignement d'ontologies est [3]. Ces travaux comparent différentes mesures de similarité pour l'alignement

4. http://oaei.ontologymatching.org/2021/knowledgegraph/index.html

Paire	Total*	LogMap	AML	PropString	PropMatch
conference-iasted	1	0.00	0.00	0.00	0.00
cmt-conference	3	**0.50**	**0.50**	**0.50**	0.33
edas-ekaw	4	0.40	0.00	0.00	**0.67**
conference-ekaw	2	**0.40**	0.00	0.00	0.00
cmt-ekaw	3	0.00	0.00	0.00	**1.00**
edas-sigkdd	4	0.00	**0.86**	0.67	0.57
cmt-confOf	6	0.00	0.00	0.29	**0.80**
confOf-sigkdd	1	0.00	0.00	**1.00**	0.67
cmt-sigkdd	2	**1.00**	**1.00**	**1.00**	**1.00**
conference-edas	3	0.67	0.00	**0.80**	**0.80**
cmt-edas	5	0.00	**0.57**	0.00	0.28
conference-sigkdd	3	**0.50**	**0.50**	**0.50**	**0.50**
confOf-edas	5	0.00	**0.80**	0.33	0.75
conference-confOf	4	0.00	0.00	**0.67**	**0.67**

TABLE 3 – Évaluation du système proposé, de l'AML et de LogMap dans le cadre de la conférence en termes de F-mesure. *Nombre d'alignements de référence.

	p			d+p			p+r			d+p+r		
Paire	P	R	F-m	P	R	F-m	P	R	F-m	P	R	F-m
starwars-swtor	0.29	0.96	0.45	0.18	0.54	0.27	0.26	0.59	0.36	0.21	0.52	0.30
malpha-stexpand	0.32	0.95	0.48	0.24	0.65	0.35	0.24	0.43	0.30	0.16	0.32	0.21
starwars-swg	0.21	1.00	0.34	0.14	0.65	0.23	0.16	0.50	0.24	0.11	0.35	0.16
mcu-marvel	0.22	0.91	0.35	0.09	0.36	0.14	0.19	0.45	0.26	0.13	0.36	0.19
malpha-mbeta	0.29	0.90	0.44	0.22	0.63	0.32	0.19	0.43	0.27	0.19	0.45	0.26

TABLE 4 – Le nombre de combinaisons de similitudes différentes dans les paires de référence du graphique de connaissances. Dans un souci d'espace, P (précision), R (rappel) et F-m (F-mesure).

	mcu-marvel			malpha-mbeta			malpha-stexpand			starwars-swg			starwars-swtor		
Paire	P	R	F-m	P	R	F-m	P	R	F-m	P	R	F-m	P	R	F-m
AMD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ATMatcher	0.91	0.91	0.91	0.98	0.92	0.95	0.95	0.95	0.95	1.00	1.00	1.00	1.00	0.98	0.99
BaselineLabel	1.00	0.36	0.53	1.00	0.34	0.51	0.97	0.68	0.80	1.00	1.00	1.00	1.00	0.98	0.99
KGMatcher	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LogMap	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LSMatch	0.82	0.82	0.82	0.62	0.58	0.60	0.62	0.61	0.62	0.72	0.65	0.68	0.88	0.79	0.83
Matcha	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PropMatch	0.13	0.36	0.19	0.19	0.45	0.26	0.16	0.32	0.21	0.11	0.35	0.16	0.21	0.52	0.30

TABLE 5 – Le résultat des systèmes qui participent à la track *Knowledge Graph*.

de chaînes, telles que Levenshtein et Jaccard, ainsi que des techniques de prétraitement telles que le stemming et l'élimination des mots vides dans différents jeux de données. Les travaux montrent également que les métriques sont moins performantes lorsqu'elles sont utilisées pour faire correspondre des propriétés plutôt que des classes. Sur la base de cette analyse, les mêmes auteurs ont développé un système appelé PropString [5] avec des performances améliorées pour la mise en correspondance des propriétés en tenant compte de la similarité entre les domaines et les portées. Une autre technique importante utilisée dans le système est l'utilisation de l'étiquetage de la partie du discours (POS) [8] pour récupérer l'entité centrale des étiquettes de propriété, c'est-à-dire l'entité composé du verbe dans l'étiquette de propriété ou du nom et de ses adjectifs associés lorsqu'aucun verbe n'est trouvé. Cette méthode atteint une précision de 100% lorsqu'elle est évaluée dans le cadre du jeu de données Conférence [4]. Cependant, la métrique utilisée dans le système n'est pas en mesure de traiter les entités conceptuelles qui sont similaires mais qui n'ont pas d'étiquettes similaires. Par exemple, la correspondance entre les propriétés *hasLocation* et *heldIn*, qui ont une sémantique similaire et une faible similarité lexicale. En outre, les domaines de ces propriétés sont *Place* et *Location*, qui peuvent être considérés comme des synonymes et présentent une faible similarité lexicale. Ces exemples montrent que les méthodes utilisées pour comparer la similarité ne sont pas suffisamment robustes pour récupérer

des correspondances avec des structures de texte plus complexes.

Dans la compétition OAEI, AML [11] est le système qui a obtenu les meilleurs résultats dans l'alignement des propriétés dans le jeu de données Conférence. Il dispose d'une méthode d'alignement spécifique pour aligner les propriétés et utilise plusieurs stratégies de correspondance de chaînes enrichies de synonymes pour mesurer la similarité entre les étiquettes des propriétés. Outre sa précision élevée de 100%, son rappel est de 41%, le système ne parvient toujours pas à récupérer certaines correspondances. Ces correspondances sont sémantiquement liés et contiennent des relations logiques qui ne peuvent pas être récupérées avec la seule métrique de similarité des chaînes de caractères. Ces problèmes posent des difficultés à l'utilisation exclusive des techniques d'alignement de chaînes et montrent la nécessité de prendre en compte le contexte de l'entité dans les mesures de similarité.

La plupart des modèles utilisés dans l'alignement d'ontologies ne peuvent toujours pas répondre aux exigences de la métrique de similarité pure, car ces plongements capturent certaines relations qui ne décrivent forcément une similarité réelle. Dans la tâche d'analyse des sentiments dans le domaine du traitement du langage naturel (NLP), par exemple, certains travaux [30, 13] ont constaté que des mots apparentés tels que "bon" et "mauvais" peuvent apparaître dans un contexte similaire mais avoir des significations sémantiques opposées. Étant donné que l'apparition dans un contexte similaire ne garantit pas la similarité entre deux entités, certains travaux [31, 10, 16] ont montré que le fait d'affiner les plongements en fonction de leurs cas d'utilisation spécifiques peut conduire à de meilleurs résultats. Par exemple, les entités *Author* et *Book* sont liées et la majorité des modèles de langage peuvent donner une forte similarité entre les deux en se basant uniquement sur la similarité des étiquettes. Mais comme il ne s'agit pas de la même entité, dans l'alignement ontologique, cette similarité peut conduire le système à classer à tort ces entités comme équivalentes.

Les modèles utilisés dans les systèmes d'alignement tels que RDF2Vec [21] dans Alod2Vec [20], OWL2Vec [6] utilisé dans une version de LogMap [7], ou encore RotatE [23] utilisé dans AMD [26], sont basés sur l'hypothèse distributionnelle qui donne une similarité élevée pour les entités qui apparaissent dans le même contexte comme décrit précédemment comme Auteur et Livre ou Bon et Mauvais et limitant la capacité de ces systèmes à trouver tous les alignements pertinents. Dans ces cas, des techniques telles que l'extension de l'alignement et la réparation des correspondances [12] peuvent être complémentaire.

6 Conclusion et travaux futurs

Dans cet article, nous avons présenté des améliorations au système PropString. Notre proposition inclut l'utilisation des plongements et de l'extension de l'alignement. Cependant, bien qu'il soit capable de produire des alignements dans différentes représentations d'ontologies, le système a encore des problèmes pour mesurer la similarité des propriétés qui ont plusieurs entités dans le domaine ou dans la portée.

Pour améliorer encore les résultats du système, le traitement de domaines et portées complexes doit être adressé. Nous voudrons également exploiter l'utilisation d'autres modèles de langages, capables de mieux exprimer la similarité sémantique. Comme les modèles plus sophistiqués ont le potentiel de mieux encoder la similarité sémantique, une partie de l'architecture du système peut migrer vers une utilisation de différents modèles de plongements, ce qui donne au système plus de flexibilité et de généralité pour l'alignement des ontologies de différents domaines.

Références

[1] Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. Recent trends in word sense disambiguation : A survey. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4330–4338. ijcai.org, 2021.

[2] Dave Binkley, Marcia Davis, Dawn Lawrie, and Christopher Morrell. To camelcase or under_score. In *2009 IEEE 17th International Conference on Program Comprehension*, pages 158–167. IEEE, 2009.

[3] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz, editors, *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, volume 8219 of *Lecture Notes in Computer Science*, pages 294–309. Springer, 2013.

[4] Michelle Cheatham and Pascal Hitzler. Conference v2.0 : An uncertain version of the OAEI conference benchmark. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig A. Knoblock, Denny Vrandecic, Paul Groth, Natasha F. Noy, Krzysztof Janowicz, and Carole A. Goble, editors, *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*, volume 8797 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 2014.

[5] Michelle Cheatham and Pascal Hitzler. The properties of property alignment. In Pavel Shvaiko, Jérôme Euzenat, Ming Mao, Ernesto Jiménez-Ruiz, Juanzi Li, and Axel Ngonga, editors, *Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014*, volume 1317 of *CEUR Workshop Proceedings*, pages 13–24. CEUR-WS.org, 2014.

[6] Jiaoyan Chen, Pan Hu, Ernesto Jiménez-Ruiz, Ole Magnus Holter, Denvar Antonyrajah, and Ian

Horrocks. Owl2vec* : embedding of OWL ontologies. *Mach. Learn.*, 110(7) :1813–1845, 2021.

[7] Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, Denvar Antonyrajah, Ali Hadian, and Jaehun Lee. Augmenting ontology alignment by semantic embedding and distant supervision. In Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Óscar Corcho, Petar Ristoski, and Mehwish Alam, editors, *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science*, pages 392–408. Springer, 2021.

[8] Alebachew Chiche and Betselot Yitagesu. Part of speech tagging : a systematic review of deep learning and machine learning approaches. *J. Big Data*, 9(1) :10, 2022.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[10] Mennatallah El-Assady, Rebecca Kehlbeck, Christopher Collins, Daniel A. Keim, and Oliver Deussen. Semantic concept spaces : Guided topic model refinement using word-embedding projections. *IEEE Trans. Vis. Comput. Graph.*, 26(1) :1001–1011, 2020.

[11] Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto. The agreementmakerlight ontology matching system. In Robert Meersman, Hervé Panetto, Tharam S. Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter De Leenheer, and Dejing Dou, editors, *On the Move to Meaningful Internet Systems : OTM 2013 Conferences - Confederated International Conferences : CoopIS, DOA-Trusted Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings*, volume 8185 of *Lecture Notes in Computer Science*, pages 527–541. Springer, 2013.

[12] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap : Logic-based and scalable ontology matching. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist, editors, *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, volume 7031 of *Lecture Notes in Computer Science*, pages 273–288. Springer, 2011.

[13] Mohammed Kasri, Marouane Birjali, Mohamed Nabil, Abderrahim Beni Hssane, Anas El-Ansari, and Mohamed El Fissaoui. Refining word embeddings with sentiment information for sentiment analysis. *J. ICT Stand.*, 10(3), 2022.

[14] Leon Knorr and Jan Portisch. Fine-tom matcher results for OAEI 2021. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, and Cássia Trojahn, editors, *Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021*, volume 3063 of *CEUR Workshop Proceedings*, pages 144–151. CEUR-WS.org, 2021.

[15] Daniel Kossack, Niklas Borg, Leon Knorr, and Jan Portisch. TOM matcher results for OAEI 2021. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, and Cássia Trojahn, editors, *Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021*, volume 3063 of *CEUR Workshop Proceedings*, pages 193–198. CEUR-WS.org, 2021.

[16] Guoxuan Li. Deepfca : Matching biomedical ontologies using formal concept analysis embedding techniques. In *ICMHI 2020 : 4th International Conference on Medical and Health Informatics, Kamakura City, Japan, August, 2020*, pages 259–265. ACM, 2020.

[17] Juhani Luotolahti, Jenna Kanerva, Veronika Laippala, Sampo Pyysalo, and Filip Ginter. Towards universal web parsebanks. In Eva Hajicová and Joakim Nivre, editors, *Proceedings of the Third International Conference on Dependency Linguistics, DepLing 2015, August 24-26 2015, Uppsala University, Uppsala, Sweden*, pages 211–220. Uppsala University, Department of Linguistics and Philology, 2015.

[18] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26 : 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[19] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

[20] Jan Portisch and Heiko Paulheim. Alod2vec matcher results for OAEI 2021. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, and

Cássia Trojahn, editors, *Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021*, volume 3063 of *CEUR Workshop Proceedings*, pages 117–123. CEUR-WS.org, 2021.

[21] Petar Ristoski and Heiko Paulheim. Rdf2vec : RDF graph embeddings for data mining. In Paul Groth, Elena Simperl, Alasdair J. G. Gray, Marta Sabou, Markus Krötzsch, Freddy Lécué, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, volume 9981 of *Lecture Notes in Computer Science*, pages 498–514, 2016.

[22] Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. With more contexts comes better performance : Contextualized sense embeddings for all-round word sense disambiguation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3528–3539. Association for Computational Linguistics, 2020.

[23] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate : Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[24] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20) :10–48550, 2017.

[25] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm : Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33 : Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[26] Zhu Wang. AMD results for OAEI 2022. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, and Cássia Trojahn, editors, *Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, held as a virtual conference, October 23, 2022*, volume 3324 of *CEUR Workshop Proceedings*, pages 145–152. CEUR-WS.org, 2022.

[27] Zhu Wang and Isabel F. Cruz. Agreementmakerdeep results for OAEI 2021. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, and Cássia Trojahn, editors, *Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021*, volume 3063 of *CEUR Workshop Proceedings*, pages 124–130. CEUR-WS.org, 2021.

[28] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers : State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics, 2020.

[29] Jifang Wu, Jianghua Lv, Haoming Guo, and Shilong Ma. Daeom : A deep attentional embedding approach for biomedical ontology matching. *Applied Sciences*, 10(21) :7909, 2020.

[30] Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xuejie Zhang. Refining word embeddings for sentiment analysis. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 534–539. Association for Computational Linguistics, 2017.

[31] Michelle Yuan, Mozhi Zhang, Benjamin Van Durme, Leah Findlater, and Jordan L. Boyd-Graber. Interactive refinement of cross-lingual word embeddings. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5984–5996. Association for Computational Linguistics, 2020.

L'Apport Mutuel de la Combinaison des Tâches d'Interconnexion de Données et d'Alignement d'Ontologies pour l'Alignement Expressif

C.K. Jradeh[1], J. David[2], O. Teste[1], C. Trojahn[1]

[1] Université Toulouse 2 Jean Jaurès, IRIT
[2] Université Grenoble Alpes, INRIA

khadija.jradeh@irit.fr, jerome.david@inria.fr,
olivier.teste@irit.fr, cassia.trojahn@irit.fr

Résumé

*Plusieurs méthodes ont été proposées pour aborder les tâches d'interconnexion de données et d'alignement d'ontologies, qui sont généralement traitées séparément. Dans cet article, nous présentons **DICAP**, un algorithme qui permet leur collaboration mutuelle. Les expériences réalisées montrent que l'ajout de relations* owl:sameAs *résultant de l'interconnexion de données permet de découvrir des correspondances ontologiques supplémentaires. De plus, la présence de correspondances ontologiques permet l'extraction de règles de liage supplémentaires et discriminantes.*

Mots-clés

Alignement d'ontologies, interconnection de données.

Abstract

Several methods have been proposed to address the tasks of data interlinking and ontology matching. However, these tasks are usually handled separately. In this work, we present an algorithm that allows for their mutual collaboration. Experiments show that the addition of owl:sameAs *resulting from the data interlinking task allows to discover additional ontology alignments. Moreover, it reveals that the existence of ontology alignments allows for the extraction of additional discriminating linking rules.*

Keywords

Ontology matching, data interlinking.

1 Introduction

La représentation des connaissances basée sur les graphes de connaissance a gagné en popularité au cours des dernières décennies. Récemment, les graphes de connaissances sont utilisés par des moteurs de recherche comme Google [7], et par d'autres entreprises comme Amazon [15]. Les graphes de connaissances RDF représentent des données structurées sous la forme d'entités et de relations entre elles. Ces relations sont représentées sous forme de triplets. Un triplet est formé d'un sujet, d'un prédicat et d'un objet. Les sujets et les prédicats sont des ressources, mais les objets peuvent être des ressources ou des littéraux. Les ressources sont identifiées par des identificateurs de ressources internationalisés (IRI [1]). Les graphes de connaissance (*Knowledge Graphs*,ou KGs) sont souvent créés indépendamment les uns des autres. Par conséquent, ils peuvent contenir différents IRI qui font référence à la même entité du monde réel mais qui ne sont pas explicitement liés par la propriété d'équivalence owl:sameAs.
La propriété owl:sameAs permet de déclarer que deux IRI différentes font référence à la même entité du monde réel. Le fait de lier ces IRI permet de compléter les graphes RDF. Le problème de liaison de différentes entités à travers différents graphes s'appelle l'interconnexion des données. De même, les ontologies sur lesquelles les graphes de connaissances s'appuient peuvent décrire des concepts et des propriétés qui sont reliées sémantiquement. Le problème de la recherche de relations entre les entités d'ontologies s'appelle l'alignement d'ontologies.
De nombreuses méthodes ont été proposées dans la littérature pour résoudre l'interconnexion des données et l'alignement d'ontologies. Cependant, ces tâches sont généralement effectuées séparément. Dans cet article, nous étudions le bénéfice mutuel de la coopération entre ces deux tâches. Plus précisément, nous étudions les questions sui-

1. Internationalised Resources Identifier

vantes dans un contexte, encore peu étudié dans la littérature, des alignements expressifs :

1. Que peut apporter l'alignement d'ontologique à la tâche d'interconnexion des données ?
2. Que peut apporter l'interconnexion des données pour la tâche de mise en correspondance des ontologies ?

Dans ce but, nous proposons l'algorithme **DICAP** qui, étant donné une paire de graphes de connaissances et leurs ontologies, permet d'effectuer à la fois l'interconnexion des données et l'alignement d'ontologies. Pour la tâche d'interconnexion des données, nous choisissons **Linkex** [4] et pour la tâche de mise en correspondance des ontologies, nous choisissons **CANARD** [23]. **Linkex** produit un ensemble de clés de liage, qui à leur tour sont utilisées pour établir des liens de type owl:sameAs entre les individus des graphes de connaissances. **CANARD** produit un ensemble de correspondances complexes entre les ontologies. **Linkex** utilisera ces correspondances pour générer un ensemble de clés de liage, chaque clé étant applicable à une paire de classes spécifique définie dans la correspondance. Nous supposons que la combinaison de **CANARD** et **Linkex** permet d'identifier un plus grand nombre de clés de liage discriminatoires, ainsi que la découverte de correspondances plus complexes.

Le reste de l'article est organisé comme suit : la section 2 résume et discute les travaux liés qui ont été proposés pour l'interconnexion des données et l'alignement d'ontologies. La section 3 donne les définitions et explique les techniques utilisées dans cet article. La section 4 décrit l'algorithme **DICAP**. La section 5 décrit les évaluations réalisées et discute leur résultats. Enfin, la section 6 résume l'article et présente les directions pour les travaux futurs.

2 Travaux liés

Différentes méthodes ont été proposées pour résoudre le problème de l'interconnexion des données [24, 16, 20, 3, 12]. Ces méthodes se répartissent en deux grandes catégories : les méthodes numériques et les méthodes logiques. Les approches numériques [24, 16] réduisent la tâche d'interconnexion des données à une tâche de calcul de similarité. Les approches logiques [20, 3, 12] définissent, quant à elles, un ensemble de règles ou d'axiomes permettant d'inférer des égalités entre individus.

Les *approches numériques* calculent la similarité entre deux entités qui appartiennent à des graphes de connaissances différents. La similarité entre deux entités est calculée par des fonctions de similarité, basées sur les valeurs des propriétés de la paire d'entités donnée. Les entités qui sont suffisamment similaires sont considérées comme identiques et sont liées par la propriété owl:sameAs. Certaines approches numériques comme Silk [24] permettent à l'utilisateur de spécifier tous les conditions que les entités doivent remplir pour être liées. D'autres, comme Limes [16] utilisent différentes méthodes pour découvrir automatiquement des spécifications supplémentaires.

Les *approches logiques* sont, quant à elles, divisées en approches basées sur les règles et sur les clés. Les premières sont basées sur des règles permettant de dériver des liens d'identité à partir des graphes d'entrée et de leurs ontologies [20, 3, 12]. Les approches basées sur les clés visent à extraire un ensemble de clés. Chaque clé est constituée d'un ensemble de propriétés et d'une classe, les propriétés permettant d'identifier de manière unique une instance qui appartient à la classe spécifiée. Selon cette définition, deux instances qui ont les mêmes valeurs pour les propriétés d'une clé sont considérées comme identiques. Plus précisément, une clé a la forme $(\{p_1, ..., p_k\}\ \mathsf{key}\ C)$ où $p_1, \cdots, p_k$ sont des propriétés et C une classe. Un exemple de clé est le suivant :

$$(\{\mathsf{creator}, \mathsf{title}\}\ \mathsf{key}\ \mathsf{Work})$$

indiquant que lorsque deux instances de la classe Work partagent respectivement les valeurs du rôle creator et du rôle title, elles désignent la même entité.

Pour utiliser des approches basées sur les clés afin de relier une paire d'ensembles de données, les clés candidates sont d'abord extraites des ensembles de données, puis les meilleures clés candidates sont sélectionnées en fonction de différentes mesures de qualité [21, 9, 2]. Lorsque les graphes RDF sont décrits à l'aide de la même ontologie, les clés peuvent être utilisées directement pour interconnecter ces graphes RDF. Par contre, pour interconnecter des graphes RDF qui sont décrits à l'aide de différentes ontologies, les clés doivent être combinées avec des alignements d'ontologies qui relient les propriétés et les classes. Ainsi pour interconnecter deux graphes RDF il est nécessaire d'avoir soit la même ontologie décrivant les deux graphes, soit un alignement entre les ontologies. Les clés de liage (Section 3.3) permettent de surmonter cette limitation.

D'autre part, les méthodes d'alignement d'ontologies sont utilisées pour trouver des corres-

pondances entre les entités de deux ontologies. Il existe différents types de méthodes d'alignement [19, 10, 11], notamment les méthodes basées sur les instances, lexicales, structurales, sémantiques, et hybrides.
Il existe deux types de correspondances, simples et complexes. Une correspondance simple fait référence à une relation sémantique de base entre deux concepts ou propriétés. Une correspondance complexe fait référence à un relation entre deux ontologies qui implique plusieurs classes ou propriétés. Il existe plusieurs méthodes pour extraire des correspondances simples [10, 17] et complexes [11, 19].
L'article [10] présente un outil appelé AgreementMakerLight AML, utilisé pour aligner automatiquement des ontologies. L'outil utilise diverses mesures pour comparer les concepts et les relations entre différentes ontologies, telles que des mesures de similarité lexicale, syntaxique et sémantique. Il utilise également des sources de connaissances externes et des techniques d'apprentissage supervisé pour améliorer le processus d'alignement. Le papier évalue l'outil en utilisant des bancs d'essai standard et montre qu'il surpasse plusieurs outils d'alignement d'ontologies de pointe. Cependant, cela ne permet pas de générer des alignements complexes. Correspondances complexes ont été identifiées dans divers domaines, comme les ontologies médicales [13].
AMLC [11] permet de générer des alignements complexes. AMLC est une version de l'AML développée pour la correspondance d'ontologies complexes. AMLC comprend une implémentation d'un algorithme de correspondance d'ontologies basé sur des règles d'association, qui effectue une extraction de motifs.
Il existe peu d'approches comme PARIS [22] qui aligne les instances, les relations et les classes. La méthode calcule la probabilité d'équivalence des instances et des propriétés des différentes ontologies considérées de manière itérative jusqu'à ce qu'elle atteigne la convergence. Ensuite, elle calcule la probabilité d'équivalence entre les classes.
DICAP, d'autre part, se concentre sur l'amélioration des résultats de **CANARD** et **Linkex**. Il fournit à **CANARD** les relations owl:sameAs nécessaires pour extraire de nouvelles correspondances. Il fournit également à **Linkex** des correspondances, ce qui lui permet d'extraire des clés de liage entre les classes équivalentes.
L'impact des relations owl:sameAs sur la tâche de correspondance d'ontologies a été étudié en [18]. Les expériences menées montrent que l'inclusion de liens owl:sameAs a un impact positif sur les approches de correspondance de schémas basées sur les instances en augmentant la distance de JACCARD [2] entre les classes de l'ontologie considérée.
Dans notre approche proposée, nous étudions d'une part l'impact de la relation owl:sameAs sur **CANARD**, qui relève des méthodes basées sur les instances pour la tâche de correspondance d'ontologies. D'autre part, nous étudions l'impact de l'utilisation de la correspondance sur l'extraction de clé de liage effectuée par **Linkex**.

3 Préliminaires

Dans cette section, nous introduisons les définitions nécessaires à la compréhension du reste de l'article. Nous décrivons également les systèmes **CANARD** (Complex Alignment Need and Abox based Relation Discovery) et **Linkex** qui sont utilisés par l'algorithme que nous proposons. **CANARD** traite la tâche de mise en correspondance d'ontologies en produisant des correspondances expressives entre une paire d'ontologies peuplées par des instances. **Linkex** est capable d'aborder la tâche d'interconnexion des données en produisant un ensemble de clés de liage, ces clés de liage représentent des axiomes permettant de relier des entités à travers une paire de graphes de connaissances différents.Nous commençons par définir une correspondance simple et les correspondances complexes.

3.1 Correspondances simples et complexes

Soient o_1 et o_2 deux ontologies. Une correspondance c est définie par un quadrulet $\langle e_{o_1}, e_{o_2}, r, n\rangle$ où e_{o_1} et e_{o_2} sont des membres de la correspondance. Ils peuvent être des expressions simples ou complexes entre les entités e_{o_1} et e_{o_2} :

1. une correspondance est dite simple, si e_{o_1} et e_{o_2} sont toutes deux des expressions simples (atomiques) ;
2. une correspondance est dite complexe, si au moins l'une des expressions e_{o_1} ou e_{o_2} est une expression complexe ;
3. r est une relation entre e_{o_1} et e_{o_2}, par exemple l'équivalence ($\equiv$), plus générale ($\sqsubseteq$), plus spécifique ($\sqsupseteq$) ;
4. une valeur n (généralement comprise entre [0,1]) peut être associée à la correspondance c pour indiquer le degré de confiance que la relation r existe entre e_{o_1} et e_{o_2}.

2. La distance de JACCARD mesure la similarité entre deux ensembles(classes). Elle est définie comme la différence entre la taille de l'intersection des classes et la taille de leur union, divisée par la taille de l'union.

La correspondance

$$\langle \text{o1:AcceptedPaper},\ \text{o2:Paper} \sqcap \exists \text{o2:hasDecision.o2:Acceptance},\ \equiv, 0.8 \rangle$$

est une correspondance complexe entre l'expression simple o1:AcceptedPaper et l'expression complexe o2:Paper ⊓ ∃o2:hasDecision.o2:Acceptance. Elle indique que le concept AcceptedPaper dans $o1$ est équivalent au concept Paper ⊓ ∃hasDecision.Acceptance dans $o2$. Le concept Paper ⊓ ∃hasDecision.Acceptance désigne les entités qui appartiennent au concept Paper et qui ont une relation hasDecision ayant la valeur Acceptance. Le degré de confiance de cette correspondance est de 0,8.

3.2 Le système CANARD

CANARD [3] est une approche d'alignement qui permet de découvrir des correspondances complexes entre des ontologies peuplées en se basant sur des questions de compétences pour l'alignement (CQAs). Les CQAs représentent les besoins en connaissances d'un utilisateur. L'approche prend en entrée une paire de KGs, source et cible, leurs ontologies et les CQAs (exprimées en SPARQL) définies par l'utilisateur. Elle renvoie en sortie un ensemble de correspondances au format EDOAL [4].
L'approche est composée par plusieurs étapes :

1. transformation des CQAs en énoncés de Logique Descriptive [6] (LD) et extraction des leurs informations lexicales ;
2. récupération des instances concernées dans le KG source ;
3. extraction des descriptions des instances liées dans les KG cibles (relation owl:sameAs) ;
4. calcul de la similarité entre les descriptions des instances sources et cibles, en conservant les triplets dont la similarité est supérieure à un seuil donné ;
5. agrégation des triplets et leur traduction en DL. Chaque correspondance est transformée en une déclaration EDOAL avec une valeur de confiance, cette valeur de confiance est calculée sur la base de la similarité entre les LD source et cible.

3. https ://gitlab.irit.fr/melodi/ontology-matching/complex/

4. EDOAL est un langage d'alignement ontologique expressif et déclaratif qui permet de représenter des alignements d'ontologies [8].

3.3 Clés de liage

Les clés de liage sont des axiomes utilisés pour générer des liens entre une paire de graphes RDF décrits à l'aide de différentes ontologies [4]. Une clé de liage entre deux graphes RDF $\mathsf{KG_1}$ et $\mathsf{KG_2}$ est une expression de la forme

$$(\{\langle P_1, Q_1\rangle, ..., \langle P_n, Q_n\rangle\} \text{ linkkey } \langle C, D\rangle)$$

où ⟨C, D⟩ est une paire de concepts appartenant respectivement à $\mathsf{KG_1}$ et $\mathsf{KG_2}$ et $\langle P_1, Q_1\rangle, ..., \langle P_n, Q_n\rangle$ est une séquence non vide de paires de propriétés où pour chaque $\langle P_i, Q_i\rangle$ dans $\{\langle P_1, Q_1\rangle, ..., \langle P_n, Q_n\rangle\}$, P_i appartient à $\mathsf{KG_1}$ et Q_i appartient à $\mathsf{KG_2}$. Elle stipule que si deux entités appartenant respectivement aux concepts C et D partagent au moins une valeur pour chaque paire de propriétés $\langle P_i, Q_i\rangle$ éventuellement multivaluées alors elles sont identiques.
Un exemple de clé de liage est :

$$(\{\langle \text{creator}, \text{auteur}\rangle, \langle \text{title}, \text{titre}\rangle\} \text{ linkkey } \langle \text{NonFiction}, \text{Essai}\rangle) \quad (1)$$

en déclarant que lorsqu'une instance de la classe NonFiction et une instance de la classe Essai, partagent des valeurs pour les rôles auteur et author, et pour les rôles title et titre, respectivement, elles désignent la même entité. Dans ce cas on dit que ces instances satisfont la condition de la clé de liage (1).
Les clés de liage peuvent être construites par des experts du domaine ou extraites automatiquement de deux ensembles de données [1, 4, 5]. Une fois obtenues, les clés de liage peuvent être transmises à un outil de génération de liens tel que [20] pour générer l'ensemble des liens d'identité.

3.4 Le système Linkex

Linkex [5] est un logiciel parmettant d'extraire des clés de liage à partir d'une paire de KG RDF source et cible. L'algorithme fonctionne comme suit : d'abord les triplets de chaque KG (source et cible) sont indexés sous la forme de o → sp (object→ subject properties). Ensuite, l'algorithme itère sur les entrées communes aux deux index pour génèrer un troisième index associant chaque paire de sujets à son ensemble maximal de propriétés pour lequels les paires de sujets partage au moins une valeur. Ce troisième index sert d'ensemble de descriptions, ou contexte formel, à partir duquel un treillis de concepts est calculé. Chaque intent du treillis de concepts résultat représente une clé de liage candidate et l'extent associé correspond à l'ensemble de liens générés par

5. https ://gitlab.inria.fr/moex/linkex/

cette clé de liage candidate. Finalement, les clés de liage candidates peuvent être filtrées grâce à des mesures d'estimation de leur qualité comme la couverture et la discriminabilité [4].
La couverture est définie comme la proportion d'instances des deux classes qui pourraient être liées par la clé de liage. La discriminabilité mesure la proximité d'une clé de liage candidate à un appariement 1 à 1. L'utilisation à la fois de la couverture et de la discriminabilité établit un équilibre entre la précision et la généralité des clés de liage candidates. A l'instar de la précision et du rappel, ces mesures peuvent être agrégées par la moyenne harmonique (hmean).

4 L'algorithme DICAP

Cette section présente **DICAP** [6], un algorithme qui vise à intégrer les pipelines d'interconnexion de données et d'alignement complexe des ontologies. L'algorithme prend en entrée une paire de graphes de connaissances KG_1 et KG_2, un ensemble de CQA et un seuil de confiance ρ.
Il fonctionne comme suit, tout d'abord, il appelle les systèmes **CANARD** et **Linkex** qui renvoient, respectivement, un ensemble d'alignements CC et de clés de liage Lks.
Ces correspondances et clés de liage sont ensuite utilisées pour saturer KG_1 et KG_2 en utilisant les algorithmes 2 et 3. Comme indiqué dans la ligne 1, l'algorithme utilise un booléen, appelé enter, initialisé à true, qui permet à l'algorithme d'entrer dans la boucle.
L'algorithme appelle également **Linkex** pour générer un ensemble de clés de liage entre la paire de classes présentes dans chaque correspondance (Lignes 6 à 8). Après, si il n'y a pas de nouvelles correspondances ou de clés de liage générées, enter est mis à false. Cela signifie que l'algorithme a atteint un état stationnaire (aucune clé de liage ou correspondance supplémentaire ne peut être extraite) et que les graphes de connaissances ne peuvent plus être enrichis. En conséquence, aucune nouvelles clés de liage ou alignements ne peuvent être extraits des graphes de connaissance.
L'algorithme 2 permet de saturer les graphes de connaissances en utilisant les correspondances. Il fonctionne simplement en ajoutant des assertions de concepts pour les individus appartenant à l'un des concepts présents dans une correspondance.
Cela permet d'extraire des clés de liage entre des classes équivalentes simples et complexes. Les assertions de classes complexes ne sont généralement pas explicitement indiquées dans les graphes de connaissances, ce qui ne permet pas d'extraire des clés de liage entre elles.

L'algorithme 3 permet de saturer les graphes de connaissances en utilisant des clés de liage. Premièrement il ajoute la relation owl:sameAs entre les individus satisfaisant la condition de clé de liage ou par les mêmes individus par transitivité. Ensuite, il ajoute de nouvelles assertions de concept et de rôle impliquées par la présence de relations owl:sameAs. Nous utilisons $s+$ pour désigner la fermeture transitive de l'individu s par rapport à la relation $\approx$ (apparaissant dans les assertions), c'est-à-dire que $s+$ est l'ensemble tel que $s \in s+$, et si c owl:sameAs b ou b owl:sameAs c avec un certain $c \in s+$ alors $b \in s+$.

L'ajout de ces liens owl:sameAs permet à **CANARD** de trouver des correspondances entre le graphe de connaissances source et cible. En l'absence de ces liens, **CANARD** ne pourra pas trouver de correspondances.

Algorithme 1 : DICAP

Input : Une paire de graphes de connaissances KG_1 et KG_2, un ensemble de CQA et un seuil de confiance ρ.
Output : Un ensemble d'alignements et un ensemble de clés de liage entre KG_1 et KG_2.

```
1  Lks← ∅, CC ← ∅, enter ← true;
2  while enter do
3  |   CC ← CANARD(KG1, KG2, CQA, ρ);
4  |   Lks ← Linkex(KG1, KG2);
5  |   Appeler l'algorithme 2 et l'algorithme 3
   |    pour saturer KG1 et KG2 en utilisant CC
   |    (KG1 devient KG1' et KG2 devient KG2');
6  |   for ⟨c1, c2, r, n⟩ ∈ CC do
7  |   |   Lks ← Linkex(KG1', KG2', c1, c2);
8  |   end
9  |   if Il n'y a pas de nouvelles clés de liage
   |    ou de correspondances générées à partir
   |    de KG1' et KG2' then
10 |   |   enter ← false;
11 |   end
12 |   KG1 ← KG1', KG2 ← KG2'
13 end
14 return CC, Lks;
```

5 Expérimentation

Dans cette section, nous décrivons d'abord l'architecture de **DICAP** et ensuite nous discutons les expériences menées.

6. https ://github.com/dace-dl-anr/DICAP

Algorithme 2 : Saturate with Correspondences

Input : Une paire de graphes de connaissances KG_1 et KG_2 et un ensemble de correspondances CC.

Output : Une paire de graphes de connaissances KG'_1 et KG'_2 telle que $\mathsf{KG}_1 \subseteq \mathsf{KG}'_1$ et $\mathsf{KG}_2 \subseteq \mathsf{KG}'_2$.

$\mathsf{KG}'_1 \leftarrow \mathsf{KG}_1$, $\mathsf{KG}'_2 \leftarrow \mathsf{KG}_2$;
for $\langle c_1, c_2, r, n\rangle \in \mathsf{CC}$ *s'il existe un individu* a *tel que* $c_1(a) \in \mathsf{KG}'_i$ *pour* $i \in \{1, 2\}$ **do**
  Ajouter $c_2(a)$ à KG'_i;
end
return $\mathsf{KG}'_1, \mathsf{KG}'_2$;

5.1 Implémentions

DICAP [7] est un logiciel open-source écrit en Java. L'architecture de DICAP est divisée en 5 modules complémentaires comme indiqué sur la Figure 1.

Le premier module est le module principal qui implémente l'algorithme 1, il utilise l'API OWL [14] pour analyser les graphes de connaissances d'entrée. Ce module est chargé d'appeler **Linkex** et **CANARD**. Il utilise les modules de saturation et les modules d'analyse.

Les modules de saturation sont respectivement responsables de la saturation des clés de liage et de la saturation des correspondances. Le module de saturation des correspondances implémente l'algorithme 2 et le module de saturation par clés de liage implémente l'algorithme 3.

Ces modules utilisent à leur tour les modules d'analyse des clés de liage et des correspondances. Les modules d'analyse utilisent respectivement les API OWL et d'alignement [8] pour analyser les clés de liage et les alignements donnés par le module principal. L'API d'alignement utilise également l'API OWL.

5.2 Résultats et discussion

Nous avons choisi un sous-ensemble du jeu de données *Conférence* de la campagne OAEI [8]. Nous considérons la paire d'ontologies peuplées edas_100 et conference_100 qui présentent les caractéristiques indiquées dans le Tableau 1 :

Dans cette expérience, nous avons lancé l'Algorithme 1 des jeux de données edas_100 et conference_100 avec une valeur de confiance de 0.7. Les résultats sont résumés dans le Tableau 2. Dans cette expérience, nous avons atteint une

7. https ://github.com/dace-dl-anr/DICAP
8. https ://oaei.ontologymatching.org/

Algorithme 3 : Saturate with Link keys

Input : Une paire de graphes de connaissances KG_1 et KG_2, un ensemble de clés de liage Lks et un seuil de confiance de ρ.

Output : Une paire de graphes de connaissances KG'_1 et KG'_2 tels que $\mathsf{KG}_1 \subseteq \mathsf{KG}'_1$ et $\mathsf{KG}_2 \subseteq \mathsf{KG}'_2$.

$\mathsf{KG}'_1 \leftarrow \mathsf{KG}_1$, $\mathsf{KG}'_2 \leftarrow \mathsf{KG}_2$;
for λ *dans* Lks *et chaque paire d'instances* $a, b \in \mathsf{KG}_i, i \in \{1, 2\}$ *satisfaisant la condition de* λ **do**
  ajouter x owl : sameAs y à KG'_i, où $x \in a^+$ et $y \in b^+$;
end
for $y \approx x \in \mathsf{KG}_i, i \in 1, 2$ *tel que* $\Sigma \cap \mathsf{KG}_i = \emptyset, \Sigma \backslash \mathsf{KG}_i \neq \emptyset$, *où* $\Sigma \in C(x), C(y), R(z, x), R(z, y)$ *pour un concept* C *ou un individu* z *et un rôle* R **do**
  ajouter Σ à KG'_i;
end
return $\mathsf{KG}'_1, \mathsf{KG}'_2$;

	edas_100	**conference_100**
Taille (MB)	21.3	33.5
Individus	10 0517	21334
Axiomes	36 1087	248831
Ass. de classe	76 993	107394
DataProp.	8988	9314
ObjectProp.	196594	94020
Prop. d'ann.	21509	16361

TABLE 1 – Caractéristiques du jeu de données.

situation stationnaire après 2 itérations. Dans l'état initial, nous avions parmi les clés de liage, la clé suivante :

$$(\{\langle \text{edas:hasName}, \text{conference:has\_a\_name}\rangle\} \text{ linkkey } \langle \text{owl:Thing}, \text{owl:Thing}\rangle)$$

Nous avions également 221 correspondances complexes et simples obtenues par canard.

La saturation de edas_100 et conference_100 avec les clés de liage initiales permet d'obtenir des relations owl:sameAs supplémentaires entre ces deux RDF KGs. En conséquence, le nombre de correspondances est passé de 221 à 341. La valeur moyenne de la confiance des correspondances obtenues a également augmenté de 0.831 à 0.878 (passant par 0,864 dans l'état intermédiaire) comme indiqué dans le Tableau 3.

Les correspondances ont été utilisées comme en-

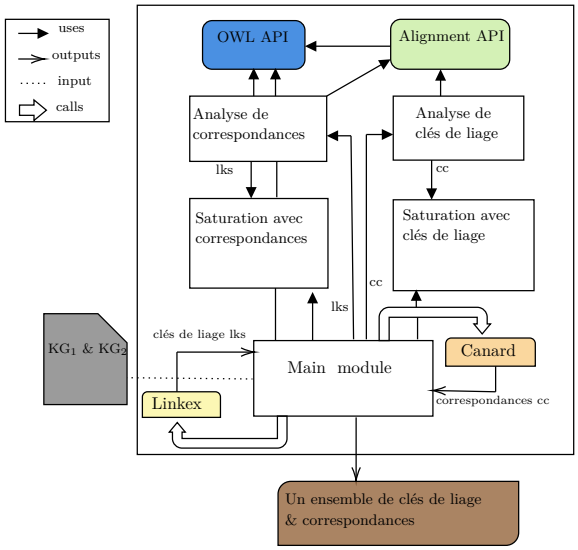


FIGURE 1 – L'architecture de **DICAP**.

	LksSEq	**LksCEq**	**Corr.**
État initial	0	0	221
État intermédiaire	23	18	291
État final	116	263	341

TABLE 2 – Le nombre de correspondances et de clés de liage entre classes simples et complexes équivalentes à différentes itérations.

trée pour **Linkex** afin d'obtenir les clés de liage suivantes :

$(\{\langle \text{rdfs:label}, \text{rdfs:label}\rangle\}$ linkkey $\langle \text{edas:Review}, \text{conference:Review}\rangle)$

$(\{\langle \text{rdfs:label}, \text{rdfs:label}\rangle\}$ linkkey $\langle \text{edas:Review}, \exists \text{conference:has\_authors.conference:Review}\rangle)$

Ces clés de liage possèdent une valeur hmean plus élevée que celles des clés originales, comme indiqué dans le Tableau 5.
Nous avons réalisé une autre expérience avec edas_100 et ekaw_100. Les résultats sont cohérents avec les résultats de l'expérience présentée.

5.3 Comparaison avec CANARD

CANARD suppose l'existence de relations owl:sameAs entre les entités des graphes de connaissances source et cible. Cependant, ce n'est pas toujours le cas. Pour cette raison, lorsque les liens d'identité ne sont pas disponibles, **CANARD** ajoute des relations synthétiques owl:sameAs entre les entités des graphes de connaissances source et cible s'ils partagent les mêmes valeurs pour la propriété rdfs:label. Cependant, cette condition est une condition faible pour imposer l'égalité entre les entités.
De façon à évaluer l'impact de l'exploitation des clé de liage sur **CANARD**, nous avons comparé premièrement le nombre de correspondances et la moyenne de leur valeur de confiance entre **DICAP**, **CANARD** (sans les égalités synthétiques, dans le Tableau 3 on y fait référence par **CANARD**$^-$) et **CANARD** (avec égalités synthétiques).
La saturation avec les clés de liage permet de générer les liens owl:sameAs entre les instances partageant les mêmes valeurs pour la propriété rdfs:label comme indiqué dans la première clé de liage de le Tableau 5a. Cette clé correspond en effet à ce que CANARD considère pour la génération de owl:sameAs dans le cas de leur absence. Les autres clés permettent de générer la propriété owl:sameAs entre les instances partageant des valeurs pour d'autres paires de propriétés présentes dans le reste des clés de liage, comme la paire de propriétés $\langle \text{hasFirstName}, \text{has\_the\_First\_Name}\rangle$ et $\langle \text{hasLastName}, \text{has\_the\_Last\_Name}\rangle$. Cela permet d'augmenter le nombre total d'entités liées par la propriété owl:sameAs, ce qui entraîne une augmentation du nombre de correspondances générées et donc une augmentation du rappel (ici, le *Couverture de CQA*) des correspondances (Tableau 4). Cependant, cette augmentation a un impact négatif sur la précision intrinsèque (Tableau 4). La couverture de CQA mesure à quel point un alignement permet de traduire un ensemble de requêtes SPARQL et la précision intrinsèque compare les instances des membres d'une correspondance. La précision intrinsèque équilibre la couverture de CQA de la manière dont la précision équilibre le rappel en recherche d'information.

5.4 Comparaison avec Linkex

Le Tableau 5 montre la hmean, la couverture et la discriminabilité des 3 principales clés de liage. Les clés de liage de la Tableau 5a sont obtenues par **Linkex** et les clés de liage du Tableau 5b sont obtenues par **DICAP**.
Les résultats représentés dans le Tableau 5 montrent que **DICAP** a pu produire de nouvelles clés de liage. Ces clés de liage possèdent une moyenne plus élevée que celle d'origine produite uniquement par **Linkex**. Les raisons derrière l'augmentation de la discriminabilité sont la présence de classes équivalentes, qui réduisent le domaine d'application de la clé de liage et per-

	DICAP	**CANARD**$^-$	**CANARD**
Nombre de corr.	341	206	221
Moyenne confidence	0.877	0.831	0.846

TABLE 3 – Comparaison entre **CANARD** et **DICAP**.

	DICAP	**CANARD**
Précision intrinsèque	0.076	0.109
Couverture de CQA	0.5	0.357

TABLE 4 – La précision et le couverture des correspondances générées par **DICAP** et **CANARD**.

mettent de générer des liens similaires à une correspondance 1 à 1. L'augmentation de la couverture est causée par la capacité de ces clés de liage à lier plus d'entités. Puisque la probabilité que les instances partagent des valeurs lorsque les entités appartiennent à des classes équivalentes est plus élevée.
Par exemple, le premier clé de liage dans le Tableau 5b

$$(\{\langle \mathsf{rdfs{:}label}, \mathsf{rdfs{:}label}\rangle)\} \text{ linkkey } \langle \mathsf{edas{:}Review}, \mathsf{conference{:}Review}\rangle) \quad (2)$$

permet de relier des entités des classes edas:Review et conference:Review si elles partagent les mêmes valeurs pour la propriété rdfs:label. Cela restreint le domaine d'application de la clé de liage en permettant de ne relier que des entités de ces classes, ce qui augmente la discriminabilité. Cela permet également de trouver plus d'instances satisfaisant la condition de cette clé de liage, car il est plus probable que les instances partagent des valeurs pour la propriété rdfs:label si elles appartiennent à des classes équivalentes. Ce n'est pas le cas pour la première clé de liage dans le Tableau 5a.

$$(\{\langle \mathsf{rdfs{:}label}, \mathsf{rdfs{:}label}\rangle)\} \text{ linkkey } \langle \mathsf{owl{:}Thing}, \mathsf{owl{:}Thing}\rangle)$$

qui relie des entités de n'importe quelle paire de classes, rendant ainsi sa correspondance loin d'être une correspondance de 1 à 1 et donc moins discriminante. La couverture de cette clé de liage est également plus faible que la clé de liage 2 car il est moins probable que les entités partagent des valeurs lorsqu'elles appartiennent à des classes non équivalentes.
Par conséquent, l'intégration de **CANARD** et **Linkex** dans **DICAP** a augmenté la qualité et le nombre de clés de liage obtenues par **Linkex** et a également augmenté le nombre de correspondances obtenues par **CANARD**.

6 Conclusion

Dans cet article, nous avons présenté **DICAP**, un algorithme combinant l'exploitation de stratégies d'interconnexion de données et d'alignement d'ontologies. Le but de cet algorithme est de tirer parti des résultats de chaque tâche pour améliorer les résultats de l'autre. Notre algorithme aborde la tâche d'interconnexion de données en utilisant des clés de liage extraites par **Linkex**. De plus, il utilise le système **CANARD** pour aborder le problème d'alignement d'ontologies en produisant des correspondances simples et complexes entre les graphes de connaissances considérés.
Nous avons mis en œuvre cet algorithme et effectué une expérience qui révèle l'importance de cet algorithme pour améliorer les résultats des systèmes considérés, c'est-à-dire **Linkex** et **CANARD**. Notamment, **DICAP** a augmenté le nombre et amélioré la discriminabilité et aussi la couverture des clés de liage générées par **Linkex**. **DICAP** a également augmenté le nombre de correspondances simples et complexes générées par **CANARD**, en augmentant en conséquence la couverture. Ainsi, notre hypothèse est valide.
Cependant, la précision de ces correspondances est inférieure à celle générée par **CANARD**. Afin d'améliorer cela, nous prévoyons d'enrichir les graphes de connaissances avec seulement des clés de liage entre des classes équivalentes. Cela permettra de créer des liens plus précis, conduisant à la génération de correspondances complexes plus précises.
Nous prévoyons également d'étendre ce travail dans plusieurs directions. Parmi elles, nous prévoyons tout d'abord de prendre en compte de nouvelles classes complexes qui n'ont pas été prises en compte dans la version actuelle, telles que celles formées à l'aide du constructeur de rôle inverse. Cela permet de saturer les graphes de connaissances avec de nouvelles assertions de classes complexes, ce qui permet d'extraire des clés de liage entre ces classes.
Nous aimerions également réaliser un ensemble d'expériences avec des jeux de données réels dé-

clé de liage	hmean	discriminabilité	couverture
{(⟨rdfs :label,rdfs :label⟩)} ⟨owl :Thing,owl :Thing⟩	0.479	0.681	0.37
{⟨hasFirstName,has_the_First_Name⟩, ⟨hasLastName,has_the_Last_Name⟩} ⟨owl :Thing,owl :Thing⟩	0.173	0.999	0.094
{⟨hasLastName,has_the_Last_Name⟩} ⟨owl :Thing,owl :Thing⟩	0.143	0.293	0.094

(a) Clés de liage obtenues par **Linkex**

clé de liage	hmean	discriminabilité	couverture
{(⟨rdfs :label,rdfs :label⟩)} ⟨edas :Review,conference :Review⟩	0.997	0.995	1
{(⟨rdfs :label,conference :has_a_name⟩)} ⟨edas :Workshop,conference :Workshop⟩	0.667	1	0.5
{(⟨rdfs :label,rdfs :label⟩)} ⟨edas :Review,∃ conference :has_authors.conference :Review⟩	0.567	0.995	0.397

(b) Clés de liage obtenues par **DICAP**

TABLE 5 – Comparaison entre les trois premiers clés de liage (selon hmean) obtenues uniquement par **Linkex** et les clés de liage obtenues par **DICAP**.

crits avec des ontologies riches telles que dbpedia et wikipedia.

Remerciements

Ce travail a été partiellement financé par le projet DACE-DL, ANR.

Références

[1] Nacira Abbas, Jérôme David, and Amedeo Napoli. Discovery of Link Keys in RDF Data Based on Pattern Structures : Preliminary Steps. In *CLA 2020 - The 15th International Conference on Concept Lattices and Their Applications*, Proceedings of the 15th International Conference on Concept Lattices and Their Applications, Tallinn / Virtual, Estonia, June 2020.

[2] Manel Achichi, Mohamed Ben Ellefi, Danai Symeonidou, and Konstantin Todorov. Automatic Key Selection for Data Linking. In *EKAW : Knowledge Engineering and Knowledge Management*, volume LNCS of *Knowledge Engineering and Knowledge Management*, pages 3–18, Bologne, Italy, November 2016. Springer International Publishing.

[3] Mustafa Al-Bakri, Manuel Atencia, Jérôme David, Steffen Lalande, and Marie-Christine Rousset. Uncertainty-sensitive reasoning for inferring sameas facts in linked data. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 698–706. IOS Press, 2016.

[4] Manuel Atencia, Jérôme David, and Jérôme Euzenat. Data interlinking through robust linkkey extraction. In *Proceedings of the Twenty-First European Conference on Artificial Intelligence*, ECAI'14, page 15–20, NLD, 2014. IOS Press.

[5] Manuel Atencia, Jérôme David, Jérôme Euzenat, Amedeo Napoli, and Jérémy Vizzini. Link key candidate extraction with relational concept analysis. *Discrete Applied Mathematics*, 273 :2–20, 2020.

[6] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, 2 edition, 2007.

[7] Dan Brickley, Matthew Burgess, and Natasha Noy. Google dataset search : Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference*, WWW '19, page 1365–1375, New York, NY, USA, 2019. Association for Computing Machinery.

[8] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment api 4.0. *Semant. Web*, 2(1) :3–10, jan 2011.

[9] Houssameddine Farah, Danai Symeonidou, and Konstantin Todorov. Keyranker : Automatic rdf key ranking for data linking. In *Proceedings of the Knowledge Capture Conference*, K-CAP 2017, New York, NY, USA, 2017. Association for Computing Machinery.

[10] Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto. The agreementmakerlight ontology matching system. In *OTM Conferences*, 2013.

[11] Daniel Faria, Catia Pesquita, Teemu Tervo, Francisco M. Couto, and Isabel F. Cruz. Aml and amlc results for oaei 2019. In *OM@ISWC*, 2019.

[12] Norbert Fuhr. Probabilistic datalog : Implementing logical information retrieval for advanced applications. *J. Am. Soc. Inf. Sci.*, 51 :95–110, 2000.

[13] Kin Wah Fung and Junchuan Xu. Synergism between the mapping projects from snomed ct to icd-10 and icd-10-cm. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2012 :218–27, 2012.

[14] Matthew Horridge and Sean Bechhofer. The owl api : A java api for working with owl 2 ontologies. In *Proceedings of the 6th International Conference on OWL : Experiences and Directions - Volume 529*, OWLED'09, page 49–58, Aachen, DEU, 2009. CEUR-WS.org.

[15] Arun Krishnan. Making search easier : How amazon's product graph is helping customers find products more easily. In *Amazon Blog*, 08 2018.

[16] Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes : A time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, page 2312–2317. AAAI Press, 2011.

[17] Lorena Otero-Cerdeira, Francisco J. Rodríguez-Martínez, and Alma Gómez-Rodríguez. Ontology matching : A literature review. *Expert Systems with Applications*, 42(2) :949–971, 2015.

[18] Joe Raad, Erman Acar, and Stefan Schlobach. On the impact of sameas on schema matching. In *Proceedings of the 10th International Conference on Knowledge Capture*, K-CAP '19, page 77–84, New York, NY, USA, 2019. Association for Computing Machinery.

[19] Marta Sabou, Elodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, and Cassia Trojahn. Survey on complex ontology matching. *Semant. Web*, 11(4) :689–727, jan 2020.

[20] Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. L2R : A Logical Method for Reference Reconciliation. In *Twenty-Second AAAI Conference on Artificial Intelligence*, page 2007, Vancouver, British Columbia, Canada, July 2007.

[21] Dezhao Song and Jeff Heflin. Automatically generating data linkages using a domain-independent candidate selection approach. In *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I*, ISWC'11, page 649–664, Berlin, Heidelberg, 2011. Springer-Verlag.

[22] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. Paris : Probabilistic alignment of relations, instances, and schema. 2011.

[23] Élodie Thiéblin, Ollivier Haemmerlé, and Cássia Trojahn. CANARD complex matching system : results of the 2018 OAEI evaluation campaign. In *OM@ISWC*, pages 138–143, 2018.

[24] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk - a link discovery framework for the web of data. In *LDOW*, 2009.

Étude de transférabilité des clés pour le liage de données entre graphes de connaissances

Thibaut Soulard[1], Fatiha Saïs[1], Joe Raad[1], Gianluca Quercini[1]

[1] LISN, CNRS (UMR 9015), Université Paris Saclay, France

Résumé

Le liage de données dans des graphes de connaissances est un problème crucial et de longue date ; il consiste à déterminer des liens entre les descriptions des entités de ces graphes désignant une même entité du monde réel. Les clés, qui sont des sous-ensembles de propriétés permettant d'identifier chaque instance d'un graphe, sont des éléments importants pour la découverte de ces liens d'identité. L'approche classique de liage de données fondée sur les clés consiste à découvrir un ensemble de clés dans chaque graphe, et ensuite appliquer une procédure de fusion (e.g., le produit cartésien des clés). Mais cette approche peut être très coûteuse en temps et peut parfois conduire à très peu de clés communes entre les deux graphes. Dans ce travail, afin de réduire le temps de calcul de la découverte de clés et par conséquent de la tâche de liage de données, nous étudions la question de transférabilité des clés découvertes dans un graphe vers un autre graphe. Plus précisément, nous avons conduit des expérimentations sur DBpedia et Wikidata afin d'évaluer l'impact en performance de la transférabilité des clés à la fois en termes de temps de calcul et de qualité des clés transférées.

Mots-clés

Liage de données, découverte de clés, graphes de connaissances, transférabilité.

Abstract

Data linking in knowledge graphs is a crucial and long-standing problem ; it involves finding links between descriptions of entities in these graphs that refer to the same real-world entity. Keys, which are subsets of properties that identify each entity, are important elements in finding these identity links. The classical key-based approach if data linking is based on the discovery of a set of keys in each graph, and then apply a merge (e.g. intersection) of these keys. But this approach can be very time consuming and can sometimes lead to very few common keys between the two graphs. In this work, in order to reduce the computational complexity of the data linking task using keys, we study the issue of transferability of keys discovered in one graph to another graph. More precisely, we conducted experiments on DBpedia and Wikidata in order to evaluate the performance impact of key transferability both in terms of computation time and quality of transferred keys.

Keywords

Data linking, key discovery, knowledge graphs, transferability.

1 Introduction

Aujourd'hui, nous assistons à une production sans précédent de ressources, publiées sous forme de données ouvertes liées. Cela conduit à la création de graphes de connaissances (KG) contenant des milliards de triplets RDF (Resource Description Framework), comme DBpedia, YAGO et Wikidata du côté académique, et le Google Knowledge Graph ou eBay Knowledge Graph du côté commercial. Ces KG contiennent des millions d'entités (telles que des personnes, des protéines ou des livres) et des millions de faits les concernant. Ces KG sont soit indépendant du domaine, comme Yago, DBpedia ou Wikidata, ou spécifiques à un domaine, comme la géographie avec Geonames [1] ou la biologie avec Bioportal [2]. En 2020, le web de données contenait plus de 650 milles graphes de données reliés entre eux. Ces derniers contiennent des connaissances qui sont généralement exprimées en RDF [3], comme des faits de la forme `<subject, propriété, object>` tels que `<Macron, presidentDe, France>`. En proposant RDF comme standard, les chercheurs de la communauté du Web sémantique ont promu la représentation des connaissances et des données sous la forme de graphes. Dans de tels graphes, les nœuds représentent des entités (par exemple, Paris) qui peuvent avoir des types représentés par des classes (par exemple, Paris est une ville), les arcs représentent des relations entre les entités (par exemple, hasMayor). Parfois, les différents types et relations sont représentés dans une ontologie OWL2 (Web Ontology Language) [4], qui définit leurs interrelations et des axiomes tels que la subsomption, la disjonction et la fonctionnalité des propriétés.

Pour pouvoir exploiter toute la richesse des données et des connaissances contenues dans ces graphes, il est important d'établir des liens sémantiques entre leurs entités. Les liens d'identité sont parmi les liens les plus importants pour l'amélioration de la complétude des KGs, puisqu'ils permettent de relier des descriptions qui réfèrent la même en-

1. https://en.wikipedia.org/wiki/GeoNames
2. https://bioportal.bioontology.org/
3. https://www.w3.org/RDF/
4. https://www.w3.org/OWL/

tité du monde réel, ce qui permet de propager les valeurs des propriétés d'une description d'entité à une autre. Différents types d'approches ont été développées pour la détection des liens d'identité dans les graphes de connaissances (voir [9, 4] pour une revue récente de la littérature).

En effet, détecter ces liens entre les entités des deux graphes G_1 et G_2 est un problème complexe, d'une part du fait de la nécessite de comparer les entités deux-à-deux ce qui donne un espace de comparaison de taille $n_1 \times n_2$ (avec n_1 et n_2 le nombre d'instances dans G_1 et G_2 respectivement).

Dans cet article, nous nous appuyons sur les travaux de liage de données fondés sur les clés de liage [11, 1] qui sont des sous-ensembles minimaux de propriétés permettant d'identifier chaque entité (e.g., l'ensemble {`nom`, `prénom`, `e-mail`} peut être une clé pour identifier les personnes). L'intérêt d'utiliser les clés pour le liage de donnée est double : (i) il permet de distinguer des sous-ensembles de propriétés ayant un fort impact sur le liage et qui pourront donc potentiellement déduire des liens d'identité avec un bon taux de précision, et (ii) il permet également de réduire le nombre de couples propriétés valeurs considérés lors de la comparaison des descriptions d'entités. Pour ce dernier point, cela peut s'avérer très efficace dans le cas où les graphes contiennent des entités qui sont décrites par un grand nombre de propriétés. En revanche, l'acquisition des clés est une tâche complexe, puisque soit celles-ci sont spécifiées par un expert du domaine ou bien découvertes automatiquement par des outils dédiés comme [15, 16, 3, 13]. Mais lorsque les graphes sont volumineux ou contiennent un grand nombre de propriétés, le coût en termes de temps de calcul de la découverte de clés, qui peut atteindre quelques jours, nécessite alors d'être réduit. C'est la raison pour laquelle, nous étudions dans ce travail s'il est possible d'économiser ce temps de calcul en appliquant un transfert de clés découvertes dans un graphe vers un autre graphe.

L'article est organisé comme suit : en section 2, nous présentons les travaux de l'état de l'art. Ensuite, en section 3 nous donnons les définitions préliminaires des notions manipulées dans ce travail. En section 4, nous décrivons la méthodologie que nous avons mise en place pour étudier la transférabilité des clés d'un graphe source à un graphe cible. Ensuite, la section 5, présente l'évaluation expérimentale, une analyse et une discussion des résultats. Enfin, en section 6 nous concluons l'article et donnons quelques perspectives.

2 État de l'art

Dès lors que nous souhaitons combiner ou exploiter des données provenant de différentes sources, il devient alors important de comparer les données et détecter les descriptions qui réfèrent la même entité du monde réel (par exemple, la même personne, le même livre, le même gène). C'est ce que nous nommons liage de données.

Diverses approches ont été proposées dans la littérature, certaines appliquent des techniques de similarité [8], d'autres utilisent l'apprentissage automatique [7, 6] ou encore des connaissances du domaine déclarées dans une ontologie [11, 1].

Une des difficultés du liage de données réside dans l'hétérogénéité des données qui rend le calcul de similarité entre descriptions d'entité plus difficile. L'autre difficulté à laquelle les outils du liage de données doivent faire face est le nombre de combinaisons de descriptions d'entités et le coût de leur comparaison. En effet, le coût de la comparaison des descriptions d'entités peut avoir un impact important sur les performances en terme de temps de calcul de la tâche de liage d'entités entre graphes de connaissances. En effet, pour décider si deux descriptions réfèrent la même entité du monde réel, une tâche de comparaison deux-à-deux des couples propriétés valeurs décrivant les entités dans G_1 et G_2 se rend nécessaire. Cette tâche est d'autant plus difficile si les propriétés sont multi-valuées.

Les *clés* sont des éléments fondamentaux pour la découverte de liens entre entités. La spécification manuelle des clés est généralement irréalisable à l'échelle du Web, en raison du volume des ensembles de données et de l'hétérogénéité de leurs entités. C'est pourquoi plusieurs approches de découverte de clés ont émergé au fil des ans, avec des applications communes dans les bases de données relationnelles [12] et les graphes de connaissances [15, 16, 3].

Pour utiliser les clés pour lier les entités représentées dans deux graphes G_1 et G_2, une approche idéale serait de découvrir les clés dans G_1 et dans G_2 et de calculer une intersection, ou d'appliquer une procédure de fusion (voir [10] pour un exemple). Mais cette approche peut être très coûteuse en temps et peut parfois conduire à très peu de clés communes entre les deux graphes. Une autre approche [3] exploite simultanément les deux graphes et l'alignement de leur propriétés pour découvrir des Linkkey qui sont valides dans les deux graphes. Tout comme les approches classiques de découverte de clés, cette approche a besoin d'exploiter les données des deux graphes.

Afin de réduire la quantité de données exploitées par la découverte de clés et par conséquent le temps de calcul de cette tâche, dans ce travail, nous étudions la question de *transférabilité* des clés découverte dans un graphe G_1 vers un autre graphe G_2. Plus précisément, est-ce-qu'il est possible de déterminer l'ensemble de clés découvertes, et donc valides dans dans un graphes G_1, qui peuvent aussi être valides dans un graphe G_2. Ainsi, il serait possible de s'affranchir de l'application de la découverte de clés dans G_2 et donc de l'étape de fusion de clés. À notre connaissance, le problème de transférabilité des clés n'a pas encore fait l'objet d'études.

3 Définitions et notions

Dans cette section nous introduisons les notions importantes utilisées dans notre approche de transfert de clés.

3.1 Graphes de connaissances

Définition 1. (Graphe de connaissances RDF). Nous considérons un graphe de connaissances défini par un couple $(\mathcal{O}, \mathcal{G})$, où :

– $\mathcal{O} = (\mathcal{C}, \mathcal{P})$ est une ontologie représentée en OWL et

composée d'un ensemble de classes $\mathcal{C}$ et de propriétés $\mathcal{P}$ pouvant être soit de type `owl:objectProperty`, dont le domaine et le co-domaine sont des classes, ou de type `owl:dataTypeProperty`, dont le domaine est une classe et le co-domaine est un type de données atomique (e.g date, string, integer).

– $\mathcal{G}$ est un ensemble de triplets RDF décrivant des instances de classes de $\mathcal{O}$ formant un graphe de données RDF.

Définition 2. (Graphe de données RDF). Un graphe de données RDF $\mathcal{G}$ est un ensemble de faits représenté par des triplets de la forme {(sujet, prédicat, objet) | sujet $\in \mathcal{I}$, propriété $\in \mathcal{P}$, objet $\in \mathcal{I} \cup \mathcal{L}$}, où $\mathcal{I}$ est l'ensemble des entités désignés des IRIs, $\mathcal{P}$ est l'ensemble des propriétés, et $\mathcal{L}$ est l'ensemble des littéraux (tels que les nombres et les chaînes de caractères).

Définition 3. (Description RDF d'une entité). Considérons une entité d'un graphe RDF $\mathcal{G}$ représentée par un IRI $i \in \mathcal{I}$, sa description RDF est l'ensemble $D(i)$ défini comme suit : $D(i) = \{(p, v) \mid (i, p, v) \in \mathcal{G}$ *ou* $(v, p, i) \in \mathcal{G}$ (pour les propriétés p de type objet)$\}$. On notera $P(i)$ l'ensemble de propriétés p tel qu'il existe une valeur v avec $(p, v) \in D(i)$. On notera $V(i, p)$ l'ensemble de valeurs v d'une propriété p apparaissant dans $D(i)$.

Définition 4. (Alignement de propriétés). Un alignement entre deux propriétés p_1 et p_2 de $\mathcal{G}_1$ et $\mathcal{G}_2$ (resp.) est une relation de mise en correspondance exprimant une relation d'équivalence sémantique entre p_1 et p_2 que nous notons par : $p_1 \equiv p_2$.

3.2 Clés

Une clé est un ensemble de propriétés permettant d'identifier de façon unique chaque instance (entité) d'une classe. Dans les graphes de connaissances, les clés peuvent être exploitées pour la détection de liens d'identité `owl:sameAs` entre descriptions d'entités dans les graphes de données RDF.

En général, si un ensemble de propriétés est déclaré comme étant une clé pour une classe, la non-satisfaction de la clé dans un graphe de données peut être due à des erreurs dans les valeurs des propriétés ou à des doublons inconnus. Alors que lorsque des paires d'instances provenant de différents graphes RDF ne satisfont pas la clé, ces paires d'instances peuvent être considérées comme des candidates à l'établissement de liens d'identité. En effet, chaque paire d'instances qui partagent les mêmes valeurs pour toutes les propriétés d'une clé peuvent être considérées comme candidates au liage d'entités.

Différentes sémantiques de clés ont été proposées dans le domaine du web sémantique (voir [2] pour une comparaison théorique et expérimentale). Elles diffèrent selon la stratégie appliquée pour gérer les propriétés multi-valuées et les valeurs non renseignées des propriétés. Dans cet article nous considérons la sémantique *S*-clé qui est celle du constructeur `owl:hasKey` [5] qui est formalisée dans la définition suivante.

Définition 5. (Sémantique d'une *S*-clé) [2]. La sémantique d'une *S*-clé $\{p_1, \ldots, p_n\}$ pour une classe [6] C est donnée dans la règle en logique du premier ordre suivante :

$$\forall x \forall y \forall z_1 ... z_n (C(x) \wedge C(y) \wedge \bigwedge_{i=1}^{n} (p_i(x, z_i) \wedge p_i(y, z_i)) \rightarrow x = y)$$

Déclarer que l'ensemble $\{p_1, \ldots, p_n\}$ est une *S*-clé pour une classe C est noté par *S*-clé$(C, (p_1, \ldots, p_n))$. Nous notons également $prop(k)$ l'ensemble de propriétés formant la clé k.

Définition 6. (Instantiation d'une clé). Soit k une *S*-clé$(C, (p_1, \ldots, p_n))$ pour une classe C dans $\mathcal{G}$. L'instantiation de la clé k pour une instance i de la classe C dans $\mathcal{G}$ est un n-uplet de valeurs de propriétés $\pi(k, i)$ définit comme suit :

$$\pi(k, i) = \{(v_1, \ldots, v_n) \mid \{(i, p_1, v_1), \ldots, (i, p_n, v_n)\} \subseteq G\}$$

4 Méthodologie de transfert de clés

Pour lier les instances d'un graphe G_1 aux instances d'un graphe G_2, l'approche classique de liage de données fondée sur les clés consiste, tout à d'abord, à découvrir un premier ensemble de clés K_1 dans le graphe G_1 et un autre ensemble de clés K_2 dans le graphe G_2, et d'appliquer une procédure de fusion de clés. La procédure de fusion peu simplement revenir au calcul de l'intersection des clés découvertes dans les deux graphes à la réécriture de propriétés équivalentes prés. Elle peut également être calculée, tel que proposé dans [10], par des produits cartésiens des ensembles de propriétés apparaissant dans K_1 et celles apparaissant dans K_2 et de conserver uniquement les minimales (une clé k_1 pour qui, il n'existe pas de clé k_2 tel que $prop(k_2) \subset prop(k_1)$. Il est important de noter que lorsque les graphes de données RDF sont décrits selon deux ontologies différentes, des alignements de propriétés (cf. définition 4) sont alors nécessaires pour réécrire les clés d'un graphe selon les propriétés alignées dans l'autre graphe (voir 4.1 la procédure d'alignement de propriétés que nous avons utilisée et la définition de la réécriture d'une clé).

Les clés obtenues de l'étape de fusion de clés sont alors considérées comme étant valides dans les deux graphes et peuvent par conséquent être utilisées pour lier les instances des deux graphes. Néanmoins, lorsque les graphes sont volumineux ou contiennent un grand nombre de propriétés le coût en termes de temps de calcul de la découverte de clés, qui peut atteindre quelques jours, nécessite alors d'être réduit. C'est la raison pour laquelle, nous avons étudié de manière expérimentale s'il est possible d'économiser ce temps de calcul en appliquant un transfert de clés découvertes dans un graphe vers un autre graphe. Pour ce faire, nous

5. `https://w3.org/TR/owl2-direct-semantics/`

6. Cela pourrait être également une expression de classe définie en OWL2 `https://www.w3.org/TR/owl2-direct-semantics/#Class_Expressions`

nous appuyons sur des mesures de qualité des clés qui sont le support et la discriminabilité que nous définissons, ci-après. Ces mesures permettent d'évaluer à quel point les clés lorsqu'elles sont transférées conservent leur propriétés d'unicité et leur capacité potentielle à produire des liens d'identité.

4.1 Alignement de propriétés

Pour calculer les alignements des propriétés décrivant les instances dans deux graphes RDF, nous considérons les ontologies auxquelles sont conformes ces graphes et avons utilisé une nouvelle approche présenté dans [14]. Cette méthode exploite les informations terminologiques et conceptuelles (i.e., définition des domaines et co-domaines des propriétés) directement disponibles dans les ontologies comme les labels, les descriptions, les types, les domaines de valeurs des propriétés ainsi que les labels et les descriptions des classes. Ensuite, à l'aide de techniques de type *transformer* telles que BERT [5] issues du traitement automatique de la langue, nous calculons des scores de similarité entre les propriétés des ontologies décrivant les deux graphes.
Cette méthode permet d'avoir un alignement $m - n$ de propriétés qui permet de capturer, notamment, des cas où les ontologies contiennent des propriétés alternatives implicites et lorsque elles diffèrent au niveau de la structure. Un exemple d'un alignement $1 - n$ judicieux pourrait être l'alignement de la propriété `wk:P625` (coordinate location) de Wikidata avec les propriétés `geo:lat` (latitude) & `geo:long` (longitude) ou bien l'alignement de `wk:P35` (head of state) avec les relations `db:monarch` & `db:leaderName` de DBpedia.

Cet alignement de propriétés peut ensuite être exploité pour réécrire les clés (voir définition 7) découvertes dans un graphe $\mathcal{G}$ en exploitant les propriétés alignées d'un graphe $\mathcal{G}'$.

Définition 7. (Réécriture d'une clé). Soient deux graphes $\mathcal{G}$ et $\mathcal{G}'$ et $\mathcal{M} = \{(p_1 \equiv p'_1), \dots (p_m \equiv p'_m)\}$ l'ensemble de m alignements de propriétés de $\mathcal{G}$ et $\mathcal{G}'$.
Soit k une S-clé$(C, (p_1, \dots, p_n))$ pour une classe C dans $\mathcal{G}$. Une réécriture $\rho(k, \mathcal{M})$ de la clé k pour le graphe $\mathcal{G}'$, selon l'ensemble des alignements de propriétés $\mathcal{M}$, est un ensemble de S-clés de la forme S-clé$(C', (p'_1, \dots, p'_n))$ tel que il existe un alignement de propriétés $m = \{(p_1 \equiv p'_1), \dots (p_n \equiv p'_n)\} \subseteq \mathcal{M}$ et que nous avons $C \equiv C'$.

4.2 Mesures de qualité des clés

Afin d'évaluer la qualité d'une clé, nous définissons ci-dessous le support d'une clé, son nombre d'exceptions et son taux de discriminabilité.

Définition 8. (Support d'une clé). Soit k une S-clé$(C, (p_1, \dots, p_n))$ pour une classe C dans $\mathcal{G}$. Le support de k dans $\mathcal{G}$ est le nombre d'instances de C ayant au moins une valeur pour chacune des propriétés $prop(k)$. Plus formellement :

$$support(k) = \mid \{x \mid \forall p \in prop(k), \exists y, (x, p, y) \in G\} \mid$$

Le support d'une clé relativement au nombre d'instances de la classe est formellement défini comme suit :

$$support^R(k) = \frac{support(k)}{\mid \{x \mid \forall x, C(x) \in G\} \mid}$$

Le support permet de mesurer la couverture d'une clé en termes de nombre d'instances pour qui la clé pourrait générer un lien d'identité. Un autre critère de qualité d'une clé est son degré de discriminabilité qui mesure à quel à point la clé permet de distinguer une instance parmi toutes les autres instances du graphe. Comme dans [3, 13], pour mesurer ce degré de discriminabilité on s'appuie sur le nombre de partitions d'instances partageant les mêmes valeurs pour les propriétés de la clé et qui sont réduites à une seule instance.

Définition 9. (Partition de l'ensemble d'instances d'une clé). Étant donnée une clé k =S-clé$(C, (p_1, \dots, p_n))$ dans un graphe de données RDF G, la partition de l'ensemble d'instances de C pouvant être formée par k est définie par l'ensemble : $\Delta(k, G) = \{\delta_1, \delta_2, \dots, \delta_m\}$. C'est l'ensemble de partitions d'instances pouvant être formées en regroupant dans chaque partition $\delta_i \in \Delta(k, G)$ le sous-ensemble d'instances partageant les mêmes valeurs pour $prop(k)$.

Définition 10. (Taux de discriminabilité d'une clé). Le taux de discriminabilité d'une clé k =S-clé$(C, (p_1, \dots, p_n))$ est le nombre de partitions $\delta_i \in \Delta(k, G)$ réduites à une seule instance relativement au nombre total de partitions de $\Delta(k, G)$. Plus formellement,

$$discr(k, G) = \frac{\mid \{\delta_i \mid \delta_i \in \Delta(k, G), \mid \delta_i \mid = 1\} \mid}{\mid \Delta(k, G) \mid}$$

Lorsque les graphes de connaissances contiennent des redondances (i.e. l'hypothèse du nom unique non vérifiée) ou contiennent des propriétés dont les valeurs sont erronées, la découverte de clés strictes devient impossible. C'est alors pour cela que des approches comme [15] ont introduit la notion de clés tolérant quelques exceptions dont le nombre est fixé par un seuil.
Ci-dessous nous donnons la définition du nombre d'exceptions d'une clé inspirée de [15].

Définition 11. (Nombre d'exceptions d'une clé). Le nombre d'exceptions d'une clé k de la forme S-clé$(C, (p_1, \dots, p_n))$ est le nombre d'instances de C qui ne satisfont pas la clé. Plus formellement :

$$ex(k) = \mid \{x \mid \forall p \in prop(k), \exists y, y \neq x, V(x, p) \cap V(y, p) \neq \emptyset\} \mid$$

La définition 11 permet de définir des clés avec des exceptions dont le nombre est calculé par rapport au

nombre d'instances de la classe dans le graphe. Cependant, cette définition n'est plus pertinente dans le cas où les valeurs des propriétés ne sont pas toutes renseignées, ce qui est souvent le cas dans les graphes de connaissances (e.g., la propriété `fax-number (P2900)` de la classe `human (Q5)` n'est renseignée que pour 31 instances dans le graphe Wikidata). Dans cet article nous proposons une nouvelle définition du nombre d'exceptions qui tient compte du support des propriétés de la clé.

Définition 12. (Nombre d'exceptions d'une clé relatif à son support). Le nombre d'exceptions d'une clé k de la forme *S*-clé$(C, (p_1, \ldots, p_n))$ relativement au nombre d'instances supportant k est formellement défini comme suit :

$$ex^R(k) = \frac{ex(k)}{support(k)}$$

4.3 Approche de transfert de clés

Afin d'étudier la transférabilité des clés d'un graphe G_1 vers un graphe G_2, nous avons procédé selon les étapes suivantes.

(1) Découverte de clés dans le graphe d'origine. Pour chaque classe d'instances du graphe G_1, pour lesquelles il existe un alignement avec une classe du graphe G_2, appliquer un outil de découverte de clés pour générer un ensemble de clés $\mathcal{K}_1$. Pour cette première étape, nous avons utilisé l'outil SAKey [15] qui découvre des *S*-clés en considérant un nombre d'exceptions passé en paramètre. Il est à noter, comme indiqué dans la section 2, SAKey étant basée sur le calcul de non-clés d'abord il ne permet pas de fournir des mesures quantitatives des clés telles que le support et la discriminabilité.

(2) Sélection des clés en fonction de leurs scores de qualité dans le graphe d'origine. Pour chaque clé découverte, nous évaluons sa qualité en termes de nombre d'exceptions générées et son support. Ces deux mesures nous permettent ensuite d'obtenir le taux d'exceptions relatif $ex^R(k)$ et de sélectionner seulement les clés qui respectent le taux d'exceptions relatif maximum ex^R_{max} passé en paramètres.

(3) Réécriture des clés sélectionnées en exploitant l'alignement de propriétés entre les deux graphes. Cette étape consiste à identifier les clés alignées, i.e., le sous-ensemble de clés $K_1 \subseteq \mathcal{K}_1$ découvertes sur G_1 pour lesquelles des alignements de propriétés existent avec le graphe G_2.

(4) Évaluation de la qualité des clés réécrites dans le graphe cible. Enfin, pour chaque réécriture de clé retenue nous évaluons sur le graphe cible G_2 sa qualité en termes de taux d'exceptions relatif $ex^R(k)$ en considérant le même seuil qu'en étape (2). Cette évaluation nous permet de détecter les clés transférées qui auraient dégénéré vers une non-clé, i.e. un ensemble de propriétés ayant un taux d'exceptions $ex^R(k) > ex^R_{max}$, et de les écarter ensuite pour les étapes de liage d'entités.

5 Évaluation expérimentale

L'évaluation expérimentale dans ce travail a pour objectif d'évaluer si le gain en performance d'une approche de liage d'entités à base de clés ayant appliqué une procédure de transfert de clés du graphe source vers un graphe cible, impacte t-il la qualité des clés dans le graphe cible. En d'autres termes, il s'agit d'étudier comment la qualité des clés évolue quand elles sont transférées à un autre graphe.
Pour répondre à cette question nous avons fait varier le taux d'exception relatif toléré pour chaque clé découverte dans le graphe source. Ceci permet de limiter l'impact de la présence d'erreurs ou de doublons dans les graphes considérés sur les résultats. Plus précisément, pour chaque taux d'exceptions maximal ex^R_{max} (avec $ex^R_{max} \in [0, 5]$), nous procédons comme suit :

1. découverte de l'ensemble de clés K_1 de C_1 dans G_1 avec chaque clé ayant un taux d'exception < ex^R_{max}
2. alignement des propriétés dans K_1 avec les propriétés dans G_2
3. utilisation T_x, un sous-ensemble des clés de K_1 dont laquelle chaque clé dans T_x a toutes ses propriétés alignées au moins à une propriété dans G_2
4. calcul du support, du taux d'exception relatif et de la discriminabilité de chaque clé dans T_x dans G_1
5. calcul du support, du taux d'exception relatif et de la discriminabilité de chaque clé dans T_x dans G_2
6. comparaison des résultats des étapes 4 et 5

5.1 Jeux de données

Les jeux de données utilisés pour l'évaluation expérimentale proviennent de DBpedia [7] et de Wikidata [8].
Pour le premier jeu de données, nous avons extrait la partie du graphe RDF représentant les instances de la classe `Person` (i.e., ayant comme `rdf:type Person`) extrait en décembre 2022. Ce premier jeu de données est ainsi composé de 1,863,013 entités et 18,960 propriétés. Cependant, il est à noter que pour un nombre non négligeable de ces propriétés correspondent à un IRI mal encodé ou erroné (IRI qui ne correspond pas à un IRI d'une des ontologies qui décrit ce graphe) comme la présence de `http://dbpedia.org/ontology/award21n,2` au lieu de `http://dbpedia.org/ontology/award`. Ces propriétés sont alors supprimées du graphe.
Le deuxième jeu de données a été extrait du graphe Wikidata du 3 mars 2021 disponible sur rdfhdt [9]. Cette extraction a été réalisée pour les IRIs typés par le triplet `<?IRI, wdt:P31, wd:Q5>` (`wd:Q5` représente la classe "Human" dans Wikidata). Ce jeu de données est composé de 3,020,916 et 3,506 propriétés.
À ces jeux de données nous avons également considéré les ontologies décrivant les instances de personnes qui contiennent respectivement 4,604 et 10,472 propriétés.

7. https://www.dbpedia.org/
8. https://www.wikidata.org/
9. https://www.rdfhdt.org/datasets/

Nous avons appliqué un algorithme d'alignement de propriétés que nous avons développé (décrit dans [14]) et qui applique une mesure de similarité fondée sur les plongements sur la description textuelle associée aux propriétés et aux classes de ces deux ontologies.

Suppression des propriétés non-pertinentes. Après l'extraction des données, une étape de pré-traitement a été appliquée pour réduire le nombre de propriétés inutilisables. Cette étape est motivée par le temps d'exécution de l'algorithme de découverte de clés qui est dépendant du nombre de propriétés dans le graphe. Pour ce faire, nous avons seulement gardé les propriétés utilisées dans les deux graphes où nous avions au moins un alignement de propriétés disponible. Nous avons gardé seulement les propriétés du type `owl:DatatypeProperty`, c'est à dire dont les valeurs sont des littéraux. Nous avons appliqué cette stratégie car une propriété non transférable ou commune aux deux graphes n'est pas utilisable pour une réécriture de clé. De plus, une clé qui est composée de `owl:objectProperty` n'est utilisable que par les outils exploitant la propagation de scores de similarité entre paires d'instances comme [1, 11]. Après ce pré-traitement, nous obtenons respectivement 239 et 135 propriétés pour DBpedia et Wikidata.

5.2 Résultats en termes de nombre de clés découvertes

Dans le tableau 1, nous présentons le nombre de Clés-Sources (C-S) pour chaque graphe de connaissances ainsi que le nombre de Clés-Réécrites (C-R) durant la vérification dans le graphe cible. Cette vérification est divisée en deux étapes, la première filtre les réécritures n'ayant aucune instantiation (cf. définition 6) (i.e. ayant un support égal à 0). Dans un second temps, nous vérifions aussi que ces réécritures respectent le taux d'exception maximal autorisé.
Le principal facteur de ce filtrage est directement liée au manque d'instantiation des réécritures dans le graphe cible. Ce manque peut être expliqué par une différence dans les données décrivant les entités, ainsi un graphe pourrait représenter les entités personnes d'un point de vue social et un autre d'un point de vue professionnel. Il est aussi explicable par un alignement $n - m$ de certaines propriétés trop lâches donnant ainsi une réécriture inutile car ces propriétés ne sont pas utilisées pour décrire un `humain` par exemple. Enfin, lors de la vérification du taux d'exceptions nous pouvons déjà avoir une première observation sur la dégénérescence des clés réécrites vers des Non-Clés Réécrites (i.e. ayant un $ex^R(k) > ex^R(k)_{max}$). Un exemple de ces Clés Réécrites et vérifiées est :
{(`wk:P2561`≡`db:name`),
(`wk:P1477`≡`db:originalName`) }

5.3 Résultats en termes d'ensemble de clés

Dans le tableau 2, nous pouvons observer l'évolution de la discriminabilité de l'ensemble des clés en fonction du taux d'exception relatif maximal. A partir de 0.5%, nous atteignons un plateau qui ne permet pas de discriminer un nombre plus important d'entités. Ce plateau est d'autant plus intéressant que pour des taux d'exceptions maximaux plus haut nous introduisons aussi plus de bruit (i.e. des exceptions) rendant la tâche final de liage d'entités plus difficile. De plus, ces données montrent également une certaine limitation de la méthode de découverte de clés avec un pourcentage de discriminabilité très faible pour les clés découvertes sur Wikidata appliquées sur ce même graphe, i.e. évaluation des mesures de qualité des clés sur le même graphe.

	ex^R_{max}	0%	0.5 %	2%	5%
DB	C-S	835	642	642	643
	C-R brut	1 572	1 199	1 199	1 200
	C-R (sup$\neq$ 0)	69	64	64	65
	C-R vérifiées	49	52	54	56
WK	C-S	357	237	238	240
	C-R brut	475	327	328	330
	C-R (sup$\neq$ 0)	82	82	82	83
	C-R vérifiées	41	48	49	55

TABLE 1 – Résultat de la découverte de clés dans DBpedia et Wikidata sur les instances représentant des personnes

	ex^R_{max}	0%	0.5 %	2%	5%
DB	DB	0.32%	80.72%	80.72%	80.72%
	WK	0.90%	28.88%	28.89%	28.89%
WK	WK	0.31%	0.88%	0.88%	1.03%
	DB	0.10%	80.73%	80.73%	80.73%

TABLE 2 – Pourcentage de discriminabilité sur l'ensemble des Clés-Réécrites découvertes dans X puis appliquées dans X & Y

5.4 Évolution du pourcentage d'exceptions après le transfert de clés

Dans la figure 1, nous montrons l'évolution du pourcentage d'exceptions en trois échelles : une réduction, une stabilité ou une augmentation du taux exceptions dans le graphe destination. Dans le cas du transfert des clés DBpedia vers Wikidata nous pouvons observer que les Clés-Réécrites ont tendance à garder leurs taux d'exceptions dans les deux graphes et donc à être relativement stable lors du transfert. Par exemple, pour les 65 clés découvertes dans DBpedia avec un taux maximale de 5%, 46 clés (70%) gardent le même taux d'exceptions quand elles sont transférées à Wikidata, 18 clés (27%) voient leur taux d'exceptions augmenter et une seule clé voit son taux d'exception diminuer. Néanmoins, dans le sens inverse nous avons un comportement bien différent avec une majorité de clés dégénérant en des Non-Clés-Réécrites. Ce comportement, impose donc à ce que les Clées-Réécrites soient bien évaluées dans le graphe cible pour s'assurer de la qualité des liens d'identité

potentiels. Enfin pour ces deux graphes le cas où la Clé-Réécrite réduit son taux d'exceptions est relativement rare.

5.5 Évolution de la discriminabilité après le transfert de clés

Dans la figure 2, nous montrons l'évolution du taux de discriminabilité d'une Clé-Réécrite en trois échelles, comme pour le pourcentage d'exceptions cas précédent : une réduction, une stabilité ou une augmentation du taux discriminabilité dans le graphe destination.
Sur ce deuxième indicateur nous observons un comportement plus régulier pour le transfert dans les deux sens, avec une majorité de Réécritures gardant un taux de discriminabilité stable.

5.6 Temps d'exécution pour chaque étape des stratégies et scénarios

Dans le tableau 3, nous présentons les différentes valeurs du temps d'exécution en fonction de la stratégie et du scénario suivi. Toutes les expériences ont été réalisé sur un CPU "Intel® Xeon® E5-2630 v4" avec 10 coeurs et 128 GB de RAM. Le point le plus important est le temps d'exécution de la découverte des Clés-Sources de DBpedia. Cette étape est le goulot d'étranglement pour la stratégie classique appliquant la fusion de clés découvertes dans les deux graphes. Par conséquent, en partant du graphe ayant le temps d'exécution le plus rapide pour la découverte de C-S, c'est à dire celui ayant le moins de propriété, (ici le graphe source serait Wikidata) nous pouvons drastiquement réduire le temps d'exécution total. Néanmoins, en partant de DBpedia nous perdons les bénéfices de cette nouvelle stratégie tout en restant tout de même dans des temps similaires à la méthode classique appliquant la fusion de clés.

5.7 Discussion

Au travers de cette évaluation expérimentale, nous avons pu analyser le comportement des clés réécrites du côté de la source et de la cible. Malgré la nécessité de vérification des clés réécrites dans le graphe cible nous avons pu montrer qu'une majorité de ces clés restent stables et ne dégénèrent pas en non clés. Néanmoins, avec la définition des exceptions relatives appliquée sur les clés trouvée par SAKey, impacte la monotonie des clés et par conséquent la complétude de l'ensemble des clés transférées. En effet, dans le tableau 4 la clé K_1 composée de $\{P_1, P_2\}$ respecte un taux maximal d'exceptions de 25% mais la clé K_2 composée de $\{P_1, P_2, P_3\}$ ne la respecte plus avec un support plus faible tout en ayant un même nombre d'exceptions donnant ainsi un taux d'exceptions de 100% alors que la clé $K_3 = \{P_1, P_2, P_3, P_4\}$ respecte ce taux.

La complétude est d'autant plus importante que des clés plus précises (i.e. ayant un support plus faible) peuvent différencier des entités que des clés plus générales (i.e. un support plus élevé) ne le peuvent pas. Un exemple de ce phénomène est illustré dans le même tableau 4, avec la clé K_1 qui ne peut pas différencier e_2 de e_3 là où K_3 le permet. Ainsi, nous pouvons espérer augmenter le taux de discriminabilité avec l'ajout de clé réécrites plus précises différenciant des cas particuliers. Cependant, cet ajout n'est pas trivial de part la perte de la monotonie des S-Clés avec le taux d'exception relatif. Mais bien-sûr cet ajout de clés engendrera une augmentation du temps d'exécution pour la découverte de clés ainsi que pour le liage d'entités, car plus de valeurs devront être comparées. Cette problématique est néanmoins commune à l'approche classique et celle appliquant le transfert de clés, ainsi elle n'a pas d'impact direct sur cet étude mais pourrait améliorer les résultats des approches basés sur les clés.

6 Conclusion et perspectives

Dans cet article, nous avons pu décrire une stratégie de transfert de clés pour éviter le recours à la découverte de clés systématique dans tous les graphes considérés lors du liage de données. Cette découverte de clés qui peut se révéler parfois très coûteuse en temps d'exécution. Nous avons aussi pu observer l'importance de l'évaluation des Clés-Réécrites dans le graphe cible pour éviter l'utilisation de Non-Clés Réécrites. Enfin, nous avons aussi pu montrer une limitation de l'utilisation des clés pour l'alignement d'entités avec un taux de discriminabilité qui peut se révéler très faible pour certain graphes.
Dans de futurs travaux, à très court termes nous envisageons d'évaluer la qualité des clés transférées en évaluant la qualité des liens d'identité (i.e. rappel, précision et F-mesure) qu'elles permettent de générer par le biais d'un outil de liage de données. Nous comptons également comparer les résultats de cette approche avec des approches qui découvrent des clés en prenant deux graphes simultanément tel que Linkkeys[3]. Enfin, nous souhaiterons développer une approche hybride pour le liage d'entités en explorant la possibilité d'utiliser les clés pour générer automatiquement un premier ensemble réduit de lien `schema:sameAs`. Ces liens pourront ensuite être utilisés pour l'initialisation des techniques d'alignement d'entités utilisant les plongements de graphes qui requiert un ensemble de liens entre les deux graphes. Nous voudrions aussi explorer la création d'un outil de découverte de clés prenant en compte les paramètres d'exception relatifs et permettant d'avoir toutes les clés respectant le taux défini.

Références

[1] Mustafa Al-Bakri, Manuel Atencia, Steffen Lalande, and Marie-Christine Rousset. Inferring same-as facts from linked data : An iterative import-by-query approach. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 9–15. AAAI Press, 2015.

[2] Manuel Atencia, Michel Chein, Madalina Croitoru, Jérôme David, Michel Leclère, Nathalie Pernelle, Fatiha Saïs, François Scharffe, and Danai Symeonidou. Defining key semantics for the RDF datasets : Expe-

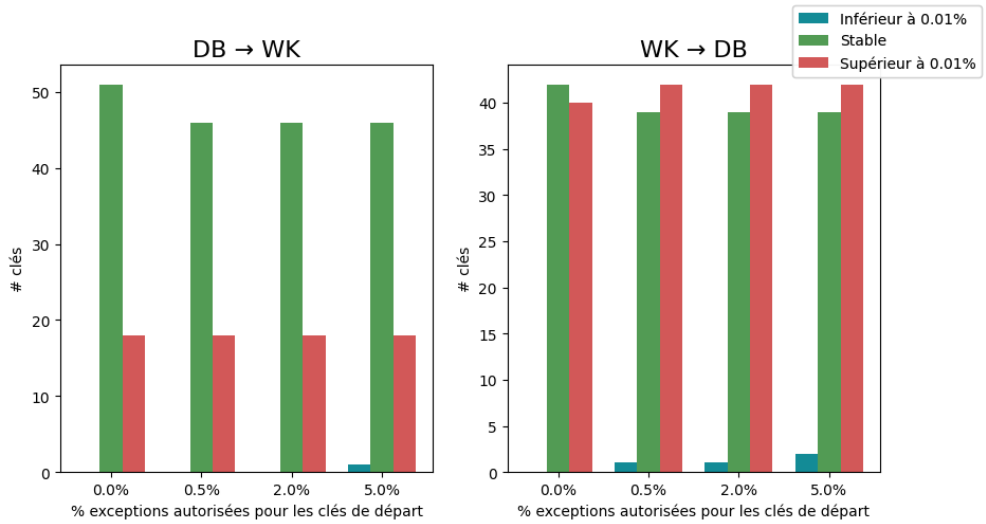


FIGURE 1 – Évolution du taux d'exceptions suite au transfert des clés découvertes

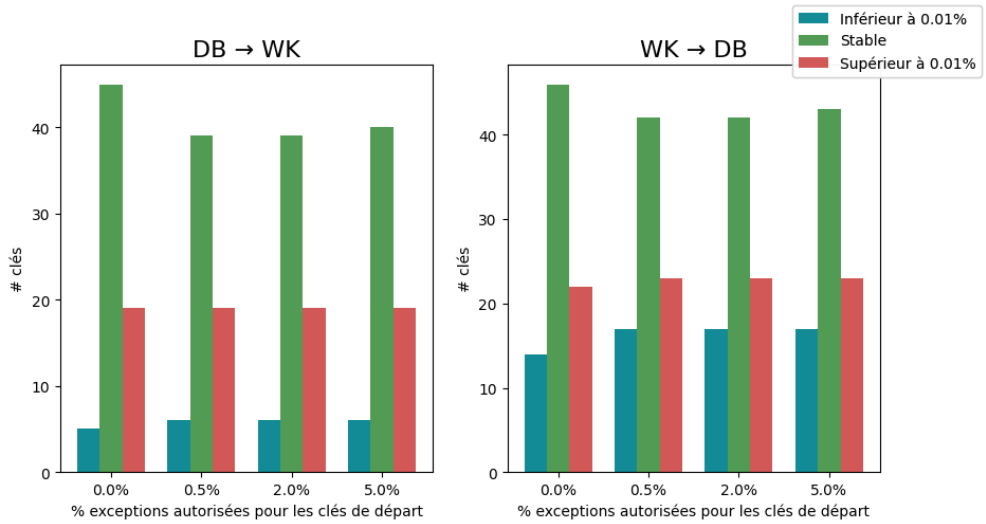


FIGURE 2 – Évolution du taux de discriminabilité suite au transfert des clés découvertes

	ex^R_{max}	0%	0.5 %	2%	5%
Classique DB ∩ WK	Découverte C-S DB	279.3	276.0	270.8	278.0
	Découverte C-S WK	5.1	5.0	5.1	5.0
	Évaluation C-S DB	33.7	25.3	25.4	25.4
	Évaluation C-S WK	3.4	1.5	1.5	1.5
	Total	321.7	308.0	302.9	310.1
Transfert Départ DB	Découverte C-S	279.3	276.0	270.8	278.0
	Évaluation C-S	33.7	25.3	25.4	25.4
	Évaluation. C-R	11.7	11.6	11.7	11.9
	Total	324.8	313.0	308.0	315.4
Transfert Départ WK	Découverte C-S	5.1	5.0	5.1	5.0
	Évaluation C-S	3.4	1.5	1.5	1.5
	Évaluation C-R	5.9	5.8	5.9	6.0
	Total	14.5	12.4	12.6	12.6

TABLE 3 – Temps d'excecution en minutes pour chaque scénario et étape.

	P_1	P_2	P_3	P_4
e_1	A1	A2	-	-
e_2	B1	B2	A3	B4
e_3	B1	B2	A3	C4
e_4	D1	D2	-	-
e_5	E1	E2	-	-

TABLE 4 – Exemple de non monotonie de clés.

riments and evaluations. In *Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iaşi, Romania, July 27-30, 2014, Proceedings*, pages 65–78, 2014.

[3] Manuel Atencia, Jérôme David, Jérôme Euzenat, Amedeo Napoli, and Jérémy Vizzini. Link key candidate extraction with relational concept analysis. *Discret. Appl. Math.*, 273 :2–20, 2020.

[4] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Comput. Surv.*, 53(6), dec 2020.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. 2018.

[6] Nikolaos Fanourakis, Vasilis Efthymiou, Dimitris Kotzinos, and Vassilis Christophides. Knowledge graph embedding methods for entity alignment : An experimental review, 2022.

[7] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, An-Hai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching : A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, page 19–34, New York, NY, USA, 2018. Association for Computing Machinery.

[8] Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes—a time-efficient approach for large-scale link discovery on the web of data. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[9] Natasha Noy, Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A survey of current link discovery frameworks. *Semant. Web*, 8(3) :419–436, jan 2017.

[10] Nathalie Pernelle, Fatiha Saïs, and Danai Symeonidou. An automatic key discovery approach for data linking. *Journal of Web Semantics*, 23 :16–30, 2013.

[11] Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. Combining a logical and a numerical method for data reconciliation. In *Journal on Data Semantics XII*, pages 66–94. Springer, 2009.

[12] Yannis Sismanis, Paul Brown, Peter J Haas, and Berthold Reinwald. Gordian : efficient and scalable discovery of composite keys. In *Proceedings of the 32nd international conference on Very large data bases*, pages 691–702, 2006.

[13] Tommaso Soru, Edgard Marx, and Axel-Cyrille Ngonga Ngomo. Rocker : A refinement operator for key discovery. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 1025–1033, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[14] Thibaut Soulard. Knowledge-based Entity Linking in Heterogeneous Knowledge Graphs at Web-Scale. Technical report, Université Paris Saclay, September 2022. Master thesis report.

[15] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. Sakey : Scalable almost key discovery in rdf data. In *International Semantic Web Conference*, pages 33–49. Springer, 2014.

[16] Danai Symeonidou, Luis Galárraga, Nathalie Pernelle, Fatiha Saïs, and Fabian Suchanek. Vickey : Mining conditional keys on knowledge bases. In *International Semantic Web Conference*, pages 661–677. Springer, 2017.

Session 4 : Ontologies et raisonnement pour les systèmes complexes

Éléments d'état de l'art sur l'extraction et la modélisation de règles formelles à partir de textes légaux

J. Bouché-Pillon[1], N. Aussenac-Gilles[1,2], P. Zaraté[1, 3], Y. Chevalier[1,4], P-Y. Gicquel[1,2]

[1] Université de Toulouse - IRIT, France
[2] CNRS
[3] Université Toulouse 1 Capitole
[4] Université Toulouse 3 Paul Sabatier

jeremy.bouche-pillon@irit.fr

Résumé

L'extraction d'énoncés formels interprétables par une machine à partir de textes en langage naturel est un champ de recherche largement étudié aujourd'hui. Un des cadres d'application de ce domaine de recherche consiste à extraire des règles formelles à partir de textes de lois et de réglementations. Cet article présente une synthèse de différentes approches d'extraction de règles à partir de réglementations légales, puis de représentation formelle de ces règles. A partir de cette synthèse, nous présentons l'ébauche d'un outil d'aide à la décision de partage d'informations sensibles entre différentes organisations, dans l'optique de valider que ces partages soient conformes aux réglementations applicables (RGPD ou directives européennes sur l'usage de l'IA).

Mots-clés

Traitement Automatique des Langues, Extraction de connaissances, Formalisme légal, Logique déontique, Représentation des connaissances, État de l'art.

Abstract

The extraction of machine-interpretable formal statements from natural language texts is a challenging and widely studied research field today. One of the application frameworks of this research field consists in extracting formal rules from legal texts and regulations. This paper presents a synthesis of different approaches to extracting rules from legal regulations, and then to formally representing these rules. From this survey, we present the draft of a tool to help of a decision support tool for sharing sensitive information between different organizations, with the aim of validating that these shares comply with the applicable regulations (GDPR or European directives on the use of AI).

Keywords

Natural Language Processing, Knowledge extraction, Legal formalism, Deontic Logic, Knowledge representation, Survey.

1 Introduction

La dénotation de la sémantique d'un texte en langue naturelle dans une langue formelle est un domaine difficile mais aussi très étudié. Avoir une extraction exacte est d'une importance primordiale en particulier lorsque l'on souhaite extraire les règles opérationnelles des lois et règlements. L'extraction de représentations formelles et en particulier de règles à partir de textes juridiques peut permettre entre autres de réguler l'accès à l'information. Cependant, la formalisation des textes de lois et réglementations demeure problématique alors qu'elle présente un enjeu majeur pour les professionnels du droit mais aussi pour les citoyens. Durant les dernière décennies, ce problème a connu de nombreuses avancées avec le développement de nouvelles approches [13, 14].

Nous nous intéressons plus particulièrement au droit des données, relatif à leur partage, leur protection ou leur traitement algorithmique par apprentissage automatique. En effet, avec l'augmentation significative des quantités et des types de données acquises, stockées, traitées et échangées par des entreprises ou organisations au cours des dernières décennies, un fort besoin de légifération a émergé afin d'encadrer leur utilisation de ces données. Plusieurs lois et réglementations ont déjà été mises en application, comme par exemple le "Réglement Général sur la Protection des Données" (RGPD)[1] à l'échelle européenne, et de nombreux autres textes seront amenés à être votés à l'avenir, comme l'"Artificial Intelligence Act" (IA Act). Toute organisation traitant des données doit se conformer et s'adapter aux nouvelles lois au fur et à mesure de leur adoption par un gouvernement, par exemple en faisant appel à un expert juridique. S'assurer que chaque traitement de données réalisé est conforme aux réglementations est une tâche laborieuse et répétitive qui nécessite des ajustements à chaque entrée en vigueur d'un nouveau texte. Le recours à des juristes s'impose alors, mais cette démarche peut ralentir le partage dans des situations où la rapidité est parfois critique. Disposer

1. General Data Protection Regulation (GDPR) – Official Legal Text, https://gdpr-info.eu/

d'un outil permettant d'automatiser l'extraction et la formalisation d'énoncés légaux en vue de les utiliser dans un système d'aide permettrait de faciliter le suivi du respect des réglementations.

Une application concrète que nous envisageons ici pour répondre à ce besoin est la conception d'un système d'aide à la décision pour des partages de données entre autorités chargées de faire respecter la loi. Ces autorités sont naturellement soumises à de nombreuses réglementations, en l'occurrence des réglementations portant sur les conditions dans lesquelles elles peuvent acquérir, traiter et partager des informations entre elles (par exemple pour obtenir des preuves en possession d'une autre autorité dans le cadre d'une enquête judiciaire). Or, la diversité de situations de partage d'informations et le flou inhérent à l'interprétation des textes légaux rendent particulièrement difficile le codage des obligations, permissions et interdictions par des informaticiens. Nous proposons ainsi une approche en 2 étapes. Dans la première, lorsqu'un texte ou une jurisprudence est identifié applicable sans ambiguïté, une décision en accord avec ce texte est prise. Dans la seconde étape, en cas d'ambiguïté, le contexte légal du partage est fourni à l'outil pour suggérer une décision qui soit en accord avec des décisions prises par le passé dans des situations similaires. Ainsi, afin de prendre en compte à la fois les textes légaux applicables et l'historique des partages, il est important de disposer d'un outil automatisant la traduction de ces textes juridiques écrits en langage naturel en règles opérationnelles utilisables par un programme de partage.
Le problème de formalisation mis en évidence précédemment se décompose en 2 étapes complémentaires, illustrées sur la figure 1 : (i) Extraire et identifier les différents éléments qui constituent les règles à partir de textes légaux. On traite ici de questionnements relatifs au domaine du Traitement Automatique des Langues. (ii) Adopter un formalisme dans lequel représenter ces règles dans l'optique de les utiliser dans un outil capable de vérifier la conformité d'un traitement de données aux réglementations en vigueur. Cette facette se rattache au domaine de la Représentation des connaissances.

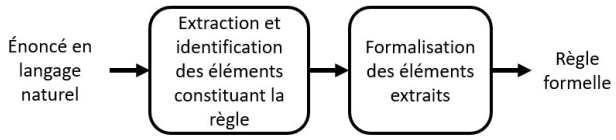


FIGURE 1 – Chaîne de traitement étudiée

A ces deux étapes, s'ajoute une phase préliminaire de caractérisation des différents éléments à extraire des textes. En effet, cette phase est indispensable pour mener une recherche précise, efficace et pertinente.
Les états de l'art récents portant sur les 3 aspects ainsi étudiés sont rares, mais on peut cependant trouver quelques états de l'art sur différents formalismes, comme [11] par exemple.
Nous structurons donc la suite de l'article en trois parties dressant un état de l'art sur la nature des éléments formant une règle juridique (2), sur les approches permettant de les extraire de textes (3) et enfin sur les formalismes définis ou utilisés pour les représenter (4). Nous esquissons en conclusion les caractéristiques d'un outil d'aide à la décision basé sur ce type de règles formalisées.

2 Les règles légales et la déontique

Afin de complètement contextualiser et caractériser les règles que nous cherchons à formaliser, nous commençons par définir ce que sont la logique et les modalités déontiques qui interviennent dans les règles légales avant de lister les différents éléments qui constituent une règle légale.

2.1 Logique et modalités déontiques

La *déontique* est une étude systématique des propriétés formelles vérifiées par des notions juridiques, comme par exemple celles de droit et d'obligation. La *logique déontique*, elle, tente ainsi de formaliser les rapports qui existent entre les caractéristiques d'une loi, les *modalités déontiques*. Il s'avère que le dénombrement de ces modalités peut être sujet à débat. En effet, dans une vision logicienne répandue héritée d'Aristote, les modalités déontiques sont au nombre de quatre, organisées en deux couples : l'interdiction et son contraire, la permission (droit de faire), d'une part, et l'obligation et son contraire, le facultatif (droit de ne pas faire), d'autre part [15]. [15] souligne cependant qu'en pratique, le "droit de faire" et le "droit de ne pas faire" se recoupent et se superposent en une unique notion à la fois négation de l'obligation et de l'interdiction : une nouvelle interprétation de "permission" qui désigne ce qui n'est ni obligatoire, ni interdit.

On considère cependant pour notre étude que la formulation des textes de loi nous permet d'effectivement distinguer "permission" et "facultatif" : on choisit de classer les formulations du style "S **peut** effectuer l'action A" comme une permission et "S **peut ne pas** effectuer l'action A" comme un facultatif.

2.2 Règles légales : composants

Nous dressons ici un bilan des différents éléments ou concepts susceptibles de se trouver dans une règle légale. Dans la littérature, les éléments généralement recherchés dans les textes en vue de la formalisation de règles légales sont traités à différents niveaux de granularité. En effet, certaines approches réalisent une analyse grammaticale et sémantique très fine dans laquelle chaque groupe de mots a un rôle et un sens précis avec des liens logiques reliant ces groupes [2, 3, 4, 7]. D'autres approches, plus générales, limitent l'analyse à l'identification de liens logiques entre groupes de mots [6]. D'autres encore se contentent d'identifier les modalités déontiques dans l'optique de classifier les règles légales [1]. Il s'avère aussi que les approches réalisant une analyse poussée identifient des éléments sensiblement différents mais en réalité complémentaires.

Ainsi, une analyse fine du contenu des règles légales permet

de repérer des éléments systématiquement présents, comme ceux mentionnés dans [3] :

- l'"agent" de la règle : Il s'agit d'un rôle sémantique rattaché au rôle grammatical de "sujet" dans une phrase. L'"agent" est ainsi l'entité qui est soumis à la règle.

- le "thème" de la règle : Analogiquement à l'"agent", le "thème" est un rôle sémantique rattaché au rôle grammatical d'"objet" dans une phrase. Le "thème" est donc l'élément sur lequel s'applique la règle.

- les termes exprimant l'une des 4 modalités déontiques : Bien que la logique déontique repose sur les 4 modalités présentées précédemment, il est assez rare que les approches repérées dans l'état de l'art s'attachent à identifier les 4 modalités. Le plus souvent, elles se focalisent sur seulement 2 ou 3 d'entre elles. Ainsi, toutes les approches étudiées ici traitent soit les modalités d'"obligation" et de "permission" [1, 3, 6], soit ces 2 modalités avec l'interdiction" en plus [2, 7], mais aucune ne s'attarde sur la distinction entre permission et facultatif.
D'une part, le fait que les "interdictions" ne soient pas traitées dans toutes les études s'explique par le fait que beaucoup de textes de lois ne contiennent effectivement pas d'"interdictions". En effet, certains textes n'"interdisent" souvent rien à proprement parler mais définissent des sanctions à certaines actions. D'autre part, lorsque des approches traitent de "permission", il s'agit de la "permission" telle que vue dans [15] et qui désigne donc ce qui n'est ni obligatoire ni interdit.
Il existe de nombreux termes permettant d'exprimer soit une obligation, soit une interdiction, soit une permission, soit un facultatif. Ces termes sont principalement mais non exclusivement des verbes comme "devoir" (obligation), "pouvoir" (facultatif ou permission selon la présence ou non d'une négation dans le "thème") ou "être interdit de" (interdiction).

- les verbes principaux : Il s'agit des verbes qui peuvent se trouver dans les textes étudiés mais qui ne sont pas rattachés à l'une des 4 modalités déontiques. Ils permettent de préciser des actions qui doivent être réalisées en conséquence de l'application de la règle ou de contextualiser la règle.

Enfin, d'autres éléments peuvent être présents dans certaines règles déontiques, tels que ceux identifiés dans [3] et dans [7] :

- des clauses d'exception : Un élément essentiel dans l'interprétation de règles déontiques est la présence de clauses d'exceptions qui spécifient des conditions particulières dans lesquelles la règle ne s'applique pas, ou s'applique différemment. L'identification de ces clauses d'exception, bien que cruciale, s'avère être une des difficultés principales rencontrées dans l'analyse de règles déontiques, notamment parce qu'elles peuvent se présenter sous la forme d'énumérations. Cet aspect est particulièrement développé dans [3].

- des phrases conditionnelles : Les phrases conditionnelles permettent d'exprimer des structures logiques de type "Si A alors B". Ce genre de phrase est donc caractérisé par deux types d'éléments : les antécédents ("A") et les conséquents ("B"). Comme pour les clauses d'exception, une des difficultés rencontrées lors de l'identification des structures conditionnelles dans les règles déontiques vient du fait que les antécédents peuvent se trouver sous la forme d'énumérations ou de listes [3, 6, 11].Bien que ces études ne considèrent les listes ou énumérations que dans les antécédents, il ne faudrait pas exclure la possibilité de rencontrer aussi des structures conditionnelles avec des conséquents sous forme d'énumérations.

- les références croisées : [6] mentionne la présence dans de nombreux paragraphes de textes de lois de références à d'autres paragraphes du même article, à d'autres articles du même texte, voire à d'autres textes dans la législation.

4. Lorsque les objets, documents ou données concernés sont déjà pertinents pour d'autres procédures, l'autorité d'exécution peut, à la demande expresse de l'autorité d'émission et après consultation de celle-ci, transférer temporairement ces éléments de preuve, à condition qu'ils soient renvoyés à l'État d'exécution dès qu'ils ne sont plus nécessaires à l'État d'émission ou à tout autre moment ou toute autre occasion convenus entre les autorités compétentes.

FIGURE 2 – Paragraphe 4 de l'article 13 de la directive 2014/41/UE

Par exemple, en analysant le paragraphe 4 de l'article 13 de la "DIRECTIVE 2014/41/UE DU PARLEMENT EUROPÉEN ET DU CONSEIL concernant la décision d'enquête européenne en matière pénale", illustré sur la figure 2, on identifie les éléments suivants :

- "L'autorité d'exécution" est l'"agent". En effet, c'est elle qui doit appliquer la règle.

- "transférer temporairement ces éléments de preuve" est le "thème". C'est ce sur quoi porte la règle. Entre autres, on peut identifier "transférer" comme un des "verbes principaux".

- "peut" est le verbe exprimant la modalité déontique. Il s'agit ici d'une permission.

- La règle s'inscrit ici dans une structure conditionnelle sous la forme "Lorsque A, B". On a donc comme antécédent "les objets ou données concernées sont déjà pertinents pour d'autres procédures, et comme conséquent "l'autorité d'exécution peut transférer temporairement ces éléments de preuve.

- Notons que d'autres propositions subordonnées permettent de contextualiser d'avantage la règle en rajoutant des conditions qui viennent ici étoffer la partie "antécédent" de la structure conditionnelle : "à la demande expresse de l'autorité d'émission et après consultation de celle-ci" et "à condition qu'ils soient renvoyés à l'État d'exécution dès qu'ils ne sont plus nécessaires à l'État d'émission ou à tout autre moment ou toute autre occasion convenus entre les autorités compétentes". On peut également remarquer dans cet exemple que l'exécution de l'action permise crée une autre obligation pour

l'État d'exécution, et que cette obligation n'est pas introduite par un verbe spécifique mais par des formulations "à condition que" et "dès que", ce qui rend l'analyse complexe.

Notons que l'exemple traité ici est en français mais que la plupart des approches étudiées ont été conçues pour analyser des textes législatifs en anglais. Par conséquent les outils utilisés dans ces approches, que ce soient les ontologies, les analyseurs syntaxiques, lexicaux ou grammaticaux, sont dédiés à l'analyse de la langue anglaise. Ce papier ayant pour but d'explorer les approches existantes et non les outils utilisés dans ces approches, toutes les approches mentionnées sont donc réalisables pour des textes dans n'importe quelle langue à condition de disposer d'outils permettant des opérations analogues dans cette langue. Les performances obtenues peuvent cependant varier étant donné que la performance globale dépend des performances individuelles des outils intervenant dans l'analyse.

3 Extraction des éléments identifiés

Une fois caractérisés, les éléments constituant les règles déontiques doivent être extraits des textes en langage naturel. Dans un premier temps, si le texte à analyser n'est pas directement accessible au format texte, mais par exemple en PDF, il est bien sûr nécessaire d'utiliser un outil de conversion, comme c'est le cas dans [2] ou [7]. Une fois le texte disponible au bon format, plusieurs types d'analyses sont possibles.

Approches à base de modèles de langue Par exemple, différents modèles dérivés de BERT sont utilisés dans [1] afin de classifier les différents énoncés d'un texte selon, entre autres, les modalités de permission et d'obligation. En amont de cette classification par modèle de langage, [1] réalise une annotation des textes avec les ontologies AkomaNtoso [2] et LegalRuleML [8] permettant de représenter respectivement la structure de textes légaux et les concepts présents dans les textes légaux. Cette approche, bien que ne permettant pas une extraction de tous les éléments essentiels mentionnés dans la section 2, offre des bons résultats de classification pour identifier quelle modalité déontique est utilisée. L'obligation et la permission étant les seules modalités traitées dans cette approche, il pourrait être intéressant de l'étendre à la modalité d'interdiction qui intervient dans de nombreux textes de loi.

Approches basées sur des chaînes de traitement plus complètes . Ainsi [3] propose une approche purement linguistique (par opposition aux approches utilisant de l'apprentissage). Après de nombreux pré-traitements lors desquels le texte analysé est divisé en sections exploitables, un grand nombre d'opérations sont réalisés via la plateforme GATE [9] qui offre pléthore d'opérations : tokenisation, fractionnement de phrases, étiquetage morpho-syntaxique, distinction des groupes verbaux et nominaux, utilisation d'un parser. L'analyse utilise le Stanford Parser [3] qui est une référence en terme d'analyse de la langue anglaise. Elle s'appuie aussi sur des nomenclatures pour identifier les termes se rattachant aux différentes modalités déontiques ainsi que les termes marquant des structures d'exceptions ou des structures conditionnelles.

Approches reposant sur une analyse sémantique des textes à l'aide d'ontologies de domaines . C'est notamment le cas dans [4] qui utilise dans un premier temps un "modèle documentaire" pour identifier quels morceaux de texte pourront être sémantiquement annotés, chaque morceau de texte étant une "unité documentaire", puis dans un second temps un "modèle sémantique" pour associer une "unité sémantique", qui peut être soit une entité d'une ontologie soit une règle sémantique, à chaque "unité documentaire" identifiée précédemment.

Approches combinant les travaux précédent , comme l'analyse à deux niveaux réalisée dans [6] :

Au niveau du document analysé dans sa globalité La structure hiérarchique du texte est identifiée et le texte se trouve ainsi découpé en sections, sous-sections, articles et phrases. En outre, certains termes spécifiques sont extraits avec leur définition grâce à une analyse lexicale. Ces termes et leur définition sont ensuite associés dans un lexique de référence qui sera réutilisé lors de la génération des règles formelles dans un souci de désambiguïsation.

Au niveau des phrases prises séparément Dans un second temps, chaque phrase peut être découpée en clauses pour réduire sa complexité puis, à partir des informations issues du niveau d'analyse précédent, un module de "Meaning Representation" permet d'identifier des prédicats logiques et leurs arguments pour créer une représentation formelle simple intermédiaire avant le formalisme final. Enfin la génération des règles dans le formalisme final s'appuie sur une ontologie regroupant les concepts qui renvoient aux modalités déontiques de permission et d'obligation. Comme mentionné pour [1], il pourrait être intéressant d'étendre cette analyse à la modalité d'interdiction.

L'approche décrite dans [7] combine également des aspects linguistiques et sémantiques dans l'analyse du texte. En effet, après un découpage du texte en phrases, une analyse sémantique permet d'abord de générer une représentation arborescente de la structure grammaticale des phrases à travers l'utilisation d'"Universal Dependencies". Cet arbre est ensuite traduit en un graphe sémantique via le framework "4lang" [4]. Enfin, de cette représentation en graphes sémantiques, une grammaire IRTG ("Interpreted Regular Tree Grammars") est utilisée pour extraire les éléments nécessaires à l'écriture de règles qui sont ensuite associés les uns avec les autres via une heuristique simple pour obtenir des règles formelles.

2. Akoma Ntoso | Akoma Ntoso Site, http://www.akomantoso.org

3. Software > Stanford Parser, https://nlp.stanford.edu/software/lex-parser.shtml

4. 4lang concept lexicon, http://hlt.sztaki.hu/resources/4lang/

Enfin, [2] propose la chaîne de traitement la plus complète, menant 2 analyses en parallèle et combinant leurs résultats à la fin. Une première étape commune aux 2 analyses consiste à reconstituer des phrases entières en exploitant la structure arborescente des textes de loi. En effet, les phrases ont tendance à se retrouver scindées en différentes propositions qui peuvent se retrouver dans une énumération. Cette reconstitution des phrases s'appuie sur une ontologie basique décrivant la structure arborescente des phrases. Après cette étape, la chaîne de traitement se scinde en 2 branches :
- *Une approche logique* applique d'abord un parser dit "CCG" pour "Combinatory Categorical Grammar". Ce parser réalise une analyse logique de chaque phrase et génère des représentations sémantiques appelées "Discourse Representation Structures" (DRS). Les DRS s'apparentent à des formules logiques du 1er ordre et permettent de trouver certaines connections sémantiques entre les mots d'une phrase. Ces représentations sont ensuite exploitées par le framework "Boxer" [5] pour extraire les relations logiques qui relient différents blocs de texte. Des régles formelles sont alors générées à partir de ces éléments.
- *Une approche syntaxique* s'en suit, qui commence par appliquer le Stanford Parser aux phrases reconstituées pour en obtenir une représentation grammaticale. Puis les phrases sont découpées en "termes", des groupes de mots qui correpsondent en général aux différentes propositions principales et relatives qui constituent chaque phrase. Chacun de ces "termes" est ensuite annoté avec des marqueurs déontiques indiquant s'il exprime une permission, une obligation, une interdiction, ou n'exprime pas de modalité déontique. Enfin, les différents "termes" sont combinés par correspondance à des structures logiques pré-définies pour générer les règles formelles.
Une dernière étape consiste à rassembler et comparer les règles issues de ces 2 branches pour obtenir un unique ensemble de règles.

4 Représentation formelle des règles

Le dernier aspect qui doit être étudié est le formalisme dans lequel représenter les règles qui ont été extraites des textes. Le formalisme choisi dépend bien sûr en grande partie de l'application envisagée pour les règles formelles générées.

Selon [12] 3 critères ont été déterminés : L'expressivité, la réutilisabilité et la compréhension des règles par l'utilisateur, afin de trouver parmi les modèles suivants : ABAC, OBAC, RBAC, OrBAC, Multi-OrBAC, KaOS, le meilleur pour modéliser des règles juridiques européennes sur le partage d'informations en matière de criminalité.

Ensuite, certaines des approches mentionnées plus tôt utilisent des représentations inspirées ou dérivées de logiques du premier ordre ou de formules en lambda calcul pour formaliser les règles. C'est le cas dans [11] où, après un état de l'art de différents langages et formats et de leur conformité à un ensemble de critères, le langage "Legal Knowledge Interchange Format" (LKIF) est présenté. Il se base sur des axiomes qui sont des formules en logique du premier ordre et sur des règles d'inférences. L'article mentionne dans son état de l'art "Rule Markup Language" (RuleML), "Semantics of Business Vocabulary and Business Rules" (SBVR), "Semantic Web Rule Language" (SWRL) et "Rule Interchange Format" (RIF) et souligne les défauts et manquements de chacun de ces langages.

On trouve aussi dans [7] une formalisation sous forme de logique déontique dyadique, qui est une logique dédiée à la représentation de structures de la forme *Il est obligatoire de faire A étant donné B* qui peut se représenter ainsi : *O(A,B)*. [7] choisit ce formalisme car il s'avère très adapté au cas particulier du texte qui y est formalisé. Cependant, cette représentation s'avère moins adaptée à des textes complexes et très descriptifs. Il est ensuite possible de développer un prouveur à partir de cette logique pour vérifier ce que l'on souhaite.
D'autres approches utilisent d'autres formes de formalisme existants comme par exemple dans [3] où le formalisme utilisé n'est pas exactement un formalisme de représentation des règles mais un formalisme d'annotation du texte. En effet, ce sont des règles JAPE ("Java Annotation Patterns Engine") [6] qui sont utilisées dans la plateforme GATE [9] pour annoter le texte de façon à identifier les différents éléments constituant une règle déontique. Bien qu'il ne soit pas ici directement question d'une représentation formelle des règles déontiques dans leur totalité, il est facile d'imaginer regrouper les annotations obtenues en une représentation formelle adaptée.

C'est également le cas de [6] qui utilise d'abord une représentation en lambda calcul comme représentation intermédiaire puis exprime ces règles en un formalisme plus lisible, ici du PCL [10], un langage conçu pour la modélisation de processus métier en intégrant des contraintes normatives. Comme avec la représentation dans [7], ce formalisme offre la possibilité de mener des raisonnements pour vérifier le respect des règles.

Il est enfin possible de définir son propre formalisme, comme c'est le cas dans [2] où les règles formelles se représentent sous la forme :
Terme1 => [Permission] NON Terme2 Ce formalisme, bien que très simple à comprendre ne semble pas être dérivé d'une logique existante. De fait, mener des raisonnements à partir de ce formalisme n'est pas trivial et nécessiterait un travail supplémentaire que nous ne détaillerons pas ici.

5 Conclusion

Nous avons présenté un état de l'art d'approches d'extraction et de modélisation de règles formelles à partir de textes de lois écrits en langage naturel. Une analyse du pro-

5. jgordon/boxer : C&C parser & Boxer, `https://github.com/jgordon/boxer`

6. JAPE : Regular Expressions over Annotations, `https://gate.ac.uk/sale/tao/splitch8.html`

blème nous a permis de le structurer selon 3 aspects essentiels : l'identification des éléments constituant des règles déontiques, l'extraction de ces éléments et des liens qui les unissent et le choix d'une représentation formelle des règles. La plupart des approches présentées sont en accord sur les différents éléments qui peuvent constituer une règle déontique, mais selon le degré de précision de l'analyse de ces approches, les éléments identifiés sont plus ou moins détaillés. En outre, selon ce degré de précision également, les approches utilisées pour extraire les éléments des textes sont de complexités variables et peuvent aussi bien être à base d'analyses linguistiques (aspects lexicaux, syntaxiques et sémantiques), structurelles, logiques, ou par modèles de langage. Enfin, il n'existe pas de représentation formelle privilégiée dans l'absolu pour les règles déontiques. L'essentiel est de choisir une représentation qui soit pertinente pour l'application que l'on souhaite en faire. Pour notre cas d'application, une représentation à base de logique du premier ordre serait pertinente.

Remerciements

Ce travail est en partie financé grâce à une bourse de thèse du MESRI attribuée à l'INP-Toulouse, et en partie grâce au projet européen STARLIGHT (oct 2021 - oct 2015), action innovante de l'appel H2020-SU-AI02 - 'Secure and resilient Artificial Intelligence technologies, tools and solutions in support of Law Enforcement and citizen protection, cybersecurity operations and prevention and protection against adversarial Artificial Intelligence'.

Références

[1] D. Liga and M. Palmirani, Transfer Learning for Deontic Rule Classification : the Case Study of the GDPR, In *International Conference on Legal Knowledge and Information Systems*, Saarbrücken 14-16 December 2022, EasyChair, 2022.

[2] M. Dragoni, S. Villata, W. Rizzi, and G. Governatori, *Combining NLP Approaches for Rule Extraction from Legal Documents,* Presented at the 1st Workshop on MIning and REasoning with Legal texts MIREL 2016, 2016. Accessed : Jan. 13, 2023. [Online]. Available :https ://hal.science/hal-01572443

[3] A. Wyner and W. Peters, On Rule Extraction from Regulations, In *Frontiers in Artificial Intelligence and Applications*, Vol. 235, Jan. 2011, doi : 10.3233/978-1-60750-981-3-113.

[4] A. Guissé, F. Lévy, and A. Nazarenko, *Un moteur sémantique pour explorer des textes réglementaires,* IC2011, 2011, Chambéry, France. pp.8. Available :https ://hal.science/hal-00707755

[5] T. Mondary, S. Després, A. Nazarenko, and S. Szulman, *Construction d'ontologies à partir de textes : la phase de conceptualisation,* Presented at the 19èmes Journées Francophones d'Ingénierie des Connaissances (IC 2008), Jun. 2008, p. 87. Available : https ://hal.science/hal-00289613

[6] G. Ferraro et al., *Automatic Extraction of Legal Norms : Evaluation of Natural Language Processing Tools,* 2020, pp. 64–81. doi : 10.1007/978-3-030-58790-1_5.

[7] G. Recski, B. Lellmann, A. Kovacs, and A. Hanbury, *Explainable Rule Extraction via Semantic Graphs*, 2021.

[8] M. Palmirani, G. Governatori, A. Rotolo, S. Tabet, H. Boley, A. Paschke, LegalRuleML : XML-Based Rules and Norms, In F. Olken, M. Palmirani, D. Sottara(eds), *Rule-Based Modeling and Computing on the Semantic Web*, RuleML, 2011, Lecture Notes in Computer Science, vol 7018, Springer, Berlin, Heidelberg, https ://doi.org/10.1007/978-3-642-24908-2_30

[9] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, GATE : an Architecture for Development of Robust HLT applications, In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp 168–175, Philadelphia, 2002, Penn., USA, Association for Computational Linguistics.

[10] G. Governatori, A. Rotolo, A Conceptually Rich Model of Business Process Compliance, In *Proceedings of the 7th Asia-Pacific Conference on Conceptual Modelling*, pp. 3–12, APCCM 2010, ACS, Brisbane, QLD, Australia, 2010.

[11] T. F. Gordon, G. Governatori, A. Rotolo, Rules and Norms : Requirements for Rule Interchange Languages in the Legal Domain, In *Rule Interchange and Applications*, G. Governatori, J. Hall, and A. Paschke, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2009, pp. 282–296, doi : 10.1007/978-3-642-04985-9_26.

[12] P.-Y. Gicquel, J. Bouché-Pillon, P. Zaraté, N. Aussenac-Gilles, Y. Chevalier, Ontologies and Rules for Access Control : a Feature Oriented Survey, In *1st WS on Collaboration in knowledge discovery and decision making : Applications to sustainable agriculture (DECISIONING 2022)*, Jun 2022, La Plata, Argentina, pp.1-12, ⟨hal-03762626⟩

[13] G. Sartor, M. Araszkiewicz, K. Atkinsonet al., Thirty years of Artificial Intelligence and Law : the second decade, In *Artificial Intelligence and Law*, vol. 30, pp.521–557, 2022, https ://doi.org/10.1007/s10506-022-09326-7

[14] S. Villata, M. Araszkiewicz, K. Ashley et al., Thirty years of artificial intelligence and law : the third decade, In *Artificial Intelligence and Law*, vol. 30, pp.561–591, 2022, https ://doi.org/10.1007/s10506-022-09327-6

[15] P. Amselek, Les fonctions normatives ou catégories modales, In *Philosophiques*, vol. 33, no. 2, pp. 391–418, Nov. 2006, doi : 10.7202/013889ar.

Construction d'un système de recommandation basé sur des contraintes via des graphes de connaissances

Ngoc Luyen Le[1,2], Marie-Hélène Abel[1], Philippe Gouspillou[2]

[1] Université de technologie de Compiègne, CNRS, Heudiasyc
(Heuristics and Diagnosis of Complex Systems), CS 60319 - 60203 Compiègne Cedex, France.
[2] Vivocaz, 8 B Rue de la Gare, 02200, Mercin-et-Vaux, France.

Résumé

Les graphes de connaissances en RDF modélisent des entités et leurs relations via des ontologies. Leur utilisation a gagné en popularité pour la modélisation de l'information, notamment dans les systèmes de recommandation où les éléments d'information et les utilisateurs sont intégrés dans ces graphes, révélant davantage de liens et de relations. Les systèmes de recommandation basés sur des contraintes exploitent une connaissance approfondie des recommandations pour identifier celles pertinentes. En les combinant avec des graphes de connaissances, nous obtenons plusieurs avantages en termes d'ensembles de contraintes. Cet article explore la construction d'un système de recommandation basé sur des contraintes avec des graphes de connaissances RDF dans le domaine de l'achat/vente de véhicules. Nos expérimentations démontrent que l'approche proposée identifie efficacement des recommandations selon les préférences des utilisateurs.

Mots-clés

Graphe de connaissances, Système de recommandation basé sur des contraintes, Ontologie

Abstract

Knowledge graphs in RDF model entities and their relations using ontologies, and have gained popularity for information modeling. In recommender systems, knowledge graphs help represent more links and relationships between users and items. Constraint-based recommender systems leverage deep recommendation knowledge to identify relevant suggestions. When combined with knowledge graphs, they offer benefits in constraint sets. This paper explores a constraint-based recommender system using RDF knowledge graphs for the vehicle purchase/sale domain. Our experiments demonstrate that the proposed approach efficiently identifies recommendations based on user preferences.

Keywords

Knowledge graph, Constraint-based Recommender System, Ontology

1 Introduction

Dans plusieurs domaines tels que les services financiers, les biens de luxe, l'immobilier ou les automobiles, les achats généralement plus coûteux sont moins fréquents que les achats de commodité. Par conséquent, faire des recommandations de ce type de produit nécessite l'obtention de plus d'informations provenant des utilisateurs tels que leurs préférences ou leurs besoins. En d'autres termes, le système de recommandation (SdR) tente de récupérer des éléments pertinents, provenant des réponses aux questions sur ses besoins et préférences, pour établir les recommandations les plus appropriées. Par conséquent, les SdRs à base de contraintes présentent une approche typique pour répondre à ce type de domaine applicatif.

Dans les SdRs à base de contraintes, l'identification des recommandations est considérée comme un processus de satisfaction des contraintes. Certaines contraintes peuvent provenir de la définition et la connaissance du domaine concerné par l'élément, l'item à considérer pour le recommandation. D'autres contraintes peuvent se définir à partir du profil de l'utilisateur comme ses préférences [7]. La combinaison de deux types de contraintes provoque une augmentation de l'espace de recherche d'un élément. Par ailleurs, cela peut conduire à la répétition du même type de contraintes pour des groupes d'utilisateurs partageant certaines caractéristiques communes. L'utilisation de graphes de connaissances RDF avec le support d'ontologies peut aider à réduire l'ensemble de contraintes en utilisant des mécanismes de raisonnement pour déduire des informations pertinentes sur les connaissances spécifiques à un domaine. Dans cet article, nous présentons notre approche pour la construction d'un système de recommandation basé sur des contraintes via des graphes de connaissances.

Le reste de cet article est organisé comme suit. Dans la section suivante, nous présentons des travaux de la littérature sur les SdRs basés sur des contraintes. La section 3 présente nos principales contributions à la construction du SdR basé sur des contraintes exploitant les graphes de connaissances RDF. Dans la section 4, une expérimentation sur le domaine de l'achat/vente de véhicules de notre approche est présentée et discutée. Pour finir, nous concluons et avançons quelques perspectives.

2 Travaux de la littérature

Les SdRs sont une application spéciale qui estime la préférence des utilisateurs pour les éléments/items et tente de recommander les éléments les plus pertinents aux utilisateurs via la récupération d'informations [15]. Les recommandations effectuées visent à aider les utilisateurs dans divers processus de prise de décision tels que la musique à écouter ou les produits à acheter. En général, les SdRs sont généralement classés en six catégories principales : les SdRs basés sur le filtrage collaboratif, les SdRs basés sur le contenu, les SdRs basés sur la démographie, les SdRs basés sur les connaissances, les SdRs sensibles au contexte et les SdRs hybrides [13].

Si la quantité de données collectées est limitée, les résultats des systèmes tels que les SdRs de filtrage collaboratif, les SdRs basés sur le contenu et les SdRs démographiques peuvent être pauvres ou ne pas couvrir complètement le spectre des combinaisons entre les utilisateurs et les éléments. En effet, ces approches peuvent rencontrer des problèmes tels que le démarrage à froid, la rareté des données, l'analyse limitée du contexte et la spécialisation excessive [1, 17]. Les SdRs basés sur la connaissance sont proposés pour résoudre ces problèmes en sollicitant explicitement les préférences des utilisateurs pour de tels éléments et en utilisant une connaissance approfondie du domaine pour calculer des recommandations pertinentes [9]. En particulier, ce type de SdR convient bien aux situations où (i) les utilisateurs souhaitent spécifier explicitement leurs exigences ; (ii) il est difficile d'obtenir des commentaires sur les éléments ; et (iii) les commentaires peuvent être obsolètes ou sensibles au temps. Par exemple, si un élément est une voiture d'occasion, les commentaires peuvent ne pas être très utiles pour calculer des recommandations car une voiture d'occasion n'est achetée qu'une seule fois.

En considérant la manière dont les utilisateurs interagissent et la base de connaissances correspondante utilisée pour ces interactions, il existe deux types de SdRs basés sur la connaissance : les SdRs basés sur les contraintes [10] et les SdRs basés sur les cas [4]. Alors que les SdRs basés sur les cas trouvent des éléments similaires en calculant et en adaptant les recommandations en fonction des cas similaires dans le passé, les SdRs basés sur les contraintes définissent un ensemble de règles/contraintes pour faire correspondre les préférences/les exigences des utilisateurs aux propriétés des éléments. Les SdRs basés sur les contraintes ont été appliqués dans différents domaines pour aider les utilisateurs à adopter les meilleures recommandations d'éléments pertinents. Dans [3,11], les auteurs ont développé des SdRs basés sur les contraintes en se basant sur l'utilisation de bases de connaissances dans le domaine du tourisme. Dans [2], l'auteur a proposé une amélioration de l'utilisation des SdRs basés sur les contraintes en utilisant la similarité des exigences des utilisateurs. L'utilisation de règles/contraintes est devenue de plus en plus populaire pour améliorer les résultats des recommandations, comme dans les applications e-commerce [5], les systèmes de simulation [14] ou les services financiers [6].

L'achat et la vente de véhicules d'occasion n'est pas aussi fréquent que d'autres produits, et chaque véhicule n'a qu'une seule transaction. En général, les préférences des utilisateurs pour leurs véhicules préférés jouent un rôle important dans la recommandation de véhicules d'occasion pertinents. Afin d'effectuer les recommandations les plus pertinentes pour ce type de transaction, nous avons choisi de construire un SdR à base de contraintes en s'appuyant sur des graphes de connaissances. Dans la section suivante, nous présenterons en détail notre approche pour ce travail.

3 Notre approche

Dans cette section, nous présentons notre approche pour la construction d'un SdR à base de contraintes s'appuyant sur un graphe de connaissances RDF. Pour illustrer notre approche, nous utilisons des ontologies du domaine de l'e-commerce liées à l'achat et à la vente de véhicules pour créer une base de connaissances.

3.1 Graphe de connaissances via RDF

La construction d'une base de connaissances pour le domaine des véhicules se compose de trois axes principaux : les propriétés des véhicules, les profils des utilisateurs-acheteurs et les interactions entre les utilisateurs-acheteurs et les véhicules. La collecte de ces informations peut être organisée et réécrite sous forme de triplets, définis formellement comme G_V = $\{a_1^v, a_2^v, ..., a_n^v\}$ où a_i^v représente un triplet RDF complet $a_i^v = \langle sujet_i, prdicat_i, objet_i \rangle$. De même, les profils d'utilisateurs qui comprennent des informations sur les utilisateurs et leurs préférences en matière de véhicules peuvent également être définis comme un ensemble de triplets RDF : $G_U = \{a_1^u, a_2^u, ..., a_m^u\}$ où a_j^v représente un triplet RDF complet. Enfin, lorsqu'un utilisateur ajoute un élément à sa liste d'éléments préférés, cela signifie que cet élément est intéressant pour l'utilisateur. Ces interactions entre les utilisateurs et les éléments sont définies comme : $RS : G_U \times G_V \times G_C \longrightarrow Interaction$ où G_U correspond à l'utilisateur, G_V désigne la description du véhicule et G_C exprime des informations contextuelles concernant l'utilisateur et l'élément lorsque l'interaction est effectuée, par exemple, les objectifs de l'utilisateur, la date, le lieu et les informations sur les ressources. Dans notre travail, nous utilisons l'ontologie développée pour la description des véhicules et des profils utilisateurs présentée dans [13].

3.2 Système de recommandation basé sur les contraintes

Après avoir défini les bases d'un graphe de connaissances RDF pour l'achat/la vente de véhicules, nous montrons dans cette section comment définir et construire un SdR basé sur des contraintes à partir de cette source de données. Dans notre travail, nous nous concentrons sur le traitement des exigences des utilisateurs à partir de leurs préférences et de leurs informations contextuelles. Tout d'abord, les préférences des utilisateurs concernant leur véhicule préféré sont considérées comme une partie des informations dans les profils d'utilisateurs. Par conséquent, les utilisateurs

doivent fournir leurs préférences relatives aux caractéristiques du véhicule qu'ils aimeraient posséder. Par exemple, plusieurs utilisateurs peuvent avoir une préférence pour la couleur *noire* ou *blanche* pour leur véhicule, ou d'autres utilisateurs veulent un *véhicule avec 7 places pour la famille*. Deuxièmement, les informations contextuelles de l'utilisateur peuvent être les situations externes. Par exemple, l'endroit où les utilisateurs vivent ou travaillent peut être un facteur important dans la sélection des types de véhicules. Par conséquent, les informations sur les préférences des utilisateurs et le contexte de l'utilisateur jouent un rôle de contraintes afin de filtrer les éléments de recommandation pertinents pour les utilisateurs.
D'autre part, nous établissons le pont entre les exigences des utilisateurs et les éléments de description des véhicules en utilisant les descriptions de véhicules et la connaissance du domaine. Tout d'abord, la description du véhicule englobe les propriétés d'un élément donné, tandis que la connaissance du domaine fournit des informations plus approfondies sur les éléments. Par exemple, lorsqu'un utilisateur déclare son profil et exprime son intérêt pour un "profil de famille", la connaissance du domaine pour les éléments de véhicule permet la recommandation de véhicules de grande taille qui ont *un nombre de places* supérieur à *trois sièges*.
La recommandation basée sur les contraintes repose sur l'exploration des relations entre les exigences de l'utilisateur et les propriétés de l'élément. La base de connaissances dans notre cas peut être considérée comme un ensemble de variables et un ensemble de contraintes. L'utilisation de ces variables et contraintes peut constituer les éléments d'un Problème de Satisfaction de Contraintes (CSP) [8, 9]. Les solutions de ce CSP permettent de trouver les recommandations les plus pertinentes dans un SdR. La tâche de calcul et de suggestion des recommandations pour un utilisateur en fonction de ses préférences est appelée une tâche de recommandation.

Definition 1 *La tâche de recommandation est définie comme un CSP($\mathcal{V}_U$,$\mathcal{V}_I$,$\mathcal{C}$), où $\mathcal{V}_U$ =$\{vu_1, vu_2, ..., vu_n\}$ désigne un ensemble de variables qui représentent les préférences de l'utilisateur, $\mathcal{V}_I$ = $\{vi_1, vi_2, ..., vi_m\}$ est un ensemble de variables qui représentent les propriétés des éléments, $\mathcal{C} = \mathcal{C}_{KB} \cup \mathcal{C}_F$ fait référence à l'ensemble des contraintes représentant les contraintes spécifiques au domaine $\mathcal{C}_{KB}$ et l'ensemble de contraintes de filtre $\mathcal{C}_F$ qui décrivent le lien entre les préférences de l'utilisateur et les éléments.*

Dans le cadre d'une application e-commerce d'achats/-ventes de véhicules, nous pouvons extraire différentes préférences utilisateur sous la forme d'un ensemble de variables pour $\mathcal{V}_U$ et les propriétés des éléments du véhicule sous la forme de l'ensemble de variables pour $\mathcal{V}_I$. En particulier, nous illustrons les ensembles de variables par un exemple simple comme suit :

+ $\mathcal{V}_U$ = $\{vu_1 : typeDeVéhicule(sedan, suv, van)$,
  $vu_2 : couleur(bleu, noir, blanc, rouge)$,
  $vu_3 : profil(utilisateurEtudiant,$
  $utilisateurParent, profilProfessionnel)$,
  $vu_4 : nombreDeSièges(entier)$,
  $vu_5 : maxKilométrage(entier)$,
  $vu_6 : marque(texte)$, $vu_7 : maxBudget(entier)\}$
+ $\mathcal{V}_I$ = $\{vi_1 : nom(texte)$, $vi_2 : prix(entier)$,
  $vi_3 : typeDeCarrosserie(texte)$,
  $vi_4 : nombreDeSièges(entier)$,
  $vi_5 : annéeDuModèle(2021, 2020, 2019, 2018)$,
  $vi_6 : marque(Peugeot, Renault, Citroen)$,
  $vi_7 : kilométrage(entier)\}$

Chaque contrainte peut être classée en $\mathcal{C}_{KB}$ ou $\mathcal{C}_F$. Alors que les contraintes $\mathcal{C}_{KB}$ sont formées à partir de la connaissance du domaine, $\mathcal{C}_F$ définit les exigences particulières de l'utilisateur sur les éléments. Nous montrons plusieurs exemples de contraintes $\mathcal{C}_{KB}$ et $\mathcal{C}_F$ dans le tableau 1.

ID	Description de la contrainte
$\mathcal{C}_{KB1}$	Une inspection technique datant de moins de 6 mois est requise pour un véhicule d'occasion de plus de 4 ans.
$\mathcal{C}_{KB2}$	Si les utilisateurs préfèrent les longs trajets, un SUV ou un Crossover peut leur convenir.
$\mathcal{C}_{F1}$	le prix de véhicule doit être inférieur ou égal au budget maximal de l'utilisateur.
$\mathcal{C}_{F2}$	le nombre de kilomètres parcourus par le véhicule doit être inférieur au kilométrage maximal imposé par l'utilisateur.
$\mathcal{C}_{F3}$	le nombre de places du véhicule doit être égal au nombre de sièges requis par l'utilisateur.
$\mathcal{C}_{F4}$	la couleur du véhicule doit être soit blanche soit bleue.

TABLE 1 – Exemple de contraintes liées aux connaissances spécifiques au domaine et aux préférences de l'utilisateur

Definition 2 *Une recommandation (une solution) pour une tâche de recommandation donnée ($\mathcal{V}_U$,$\mathcal{V}_I$,$\mathcal{C}$) est définie comme une instanciation de $\mathcal{V}_I$ en réalisant une affectation complète aux variables de ($\mathcal{V}_U$,$\mathcal{V}_I$) telle que les contraintes en $\mathcal{C}$ soient satisfaites. La recommandation est cohérente si les affectations sont cohérentes avec les contraintes.*

Les SdRs basés sur des contraintes reposent sur une base de connaissances explicite du domaine des utilisateurs et des éléments. Avec deux types de contraintes, nous pouvons calculer des recommandations pertinentes pour un utilisateur. Les contraintes de $\mathcal{C}_{KB}$ liées à la connaissance spécifique du domaine peuvent être satisfaites en utilisant des règles qui sont intégrées dans des ontologies. Par conséquent, nous explorons cette approche dans la prochaine section en se basant sur le modèle d'ontologie repris de [12,13] et le graphe de connaissances RDF pour les profils d'utilisateurs et les descriptions de véhicules.

3.3 Contraintes de connaissance spécifiques au domaine par des règles SWRL

Dans le contexte du domaine de l'achat/vente de véhicules, des ontologies sont utilisées pour structurer et organiser les

descriptions de véhicules et les profils d'utilisateurs. L'ontologie proposée est construite à l'aide du langage d'ontologie Web (OWL) [13, 16], qui est un langage de représentation des connaissances hautement expressif, flexible et efficace basé sur l'arrière-plan mathématique de la logique de description. OWL peut réaliser le raisonnement sur les informations implicites en traitant des connaissances explicites, ce qui améliore la gestion de l'information. Les règles sont utiles pour implémenter la partie déductive de la base de connaissances. Dans ce travail, nous utilisons le langage de règles du web sémantique (SWRL) pour écrire des règles sur des graphes de connaissances RDF.

Les contraintes dans l'ensemble de contraintes $\mathcal{C}_{KB}$ s'appliquent souvent à une classe, aux propriétés d'une classe ou à un groupe d'individus. En d'autres termes, ces contraintes affectent les informations globales dans le cadre de la base de connaissances. Ces contraintes peuvent être traduites en règles à intégrer dans l'ontologie à l'aide de SWRL. Par exemple, pour la contrainte $\mathcal{C}_{KB2}$ qui est utilisée pour tous les utilisateurs ayant une préférence pour la *route longue distance*, nous pouvons utiliser une règle SWRL pour déduire le *type de véhicule* préféré par l'utilisateur. Par conséquent, nous proposons de représenter les contraintes de connaissances spécifiques au domaine à l'aide de règles SWRL, en se basant sur les avantages en matière de déduction d'informations.

ID	Expression de règle SWRL
$\mathcal{C}_{KB1}$	$Automobile(?a) \wedge ContrôleTechnique(?c) \wedge inspecté(?a, ?c) \wedge DateDeProduction(?a, ?pdate) \wedge valideDe(?c, ?cdate) \wedge temporal : duration(?pdurée, ?pdate, \text{“}maintenant\text{”}, \text{“}mois\text{”}) \wedge temporal : duration(?cdurée, ?cdate, \text{“}maintenant\text{”}, \text{“}mois\text{”}) \wedge swrlb : greaterThan(?pdurée, 48) \wedge swrlb : greaterThan(?cdurée, 6) \longrightarrow estRequis(?c, vrai)$
$\mathcal{C}_{KB2}$	$PréférenceDeVéhicule(?vpu) \wedge aLeTypeDeRoutePréféré(?vpu, ?route) \wedge sameAs(?route, upo : longDistanceRoute) \longrightarrow aUnTypeDeVéhiculePréféré(?vpu, upo : SUV) \wedge aUnTypeDeVéhiculePréféré(?vpu, upo : Crossover)$

TABLE 2 – Les règles SWRL pour les contraintes définies dans le tableau 1.

Les règles SWRL offrent de puissantes capacités déductives exploitant une modélisation ontologique. Cependant, SWRL est essentiellement un langage de règles, et il ne fournit pas de support solide pour filtrer et interroger les informations du graphe de connaissances RDF. Par conséquent, nous présenterons une approche pour les contraintes $\mathcal{C}_F$ liées aux préférences de l'utilisateur qui implique le filtrage et la mise en correspondance sur les graphes de connaissances RDF dans la section suivante.

3.4 Contraintes de préférence de l'utilisateur par des requêtes SPARQL

Supposons que Q est une requête SPARQL et que c est une contrainte. Q FILTER c est appelée une requête de contrainte, où chaque variable dans la contrainte est satisfaite dans la requête Q. Une solution d'une requête SPARQL Q est définie comme une assignation de variables dans Q à des valeurs. Un ensemble de valeurs possibles qui peuvent être assignées à une variable est appelé un domaine. Une recommandation ou solution est cohérente si toutes les variables déclarées dans la requête ont une valeur correspondante garantie. Pour trouver toutes les solutions possibles, nous sélectionnons une valeur dans le graphe de connaissances RDF pour chaque variable et nous assurons qu'elle vérifie les conditions des motifs et des filtres. En tenant compte de ces aspects, trouver des recommandations pour le SdR basé sur des contraintes défini dans les définitions 1 et 2 revient à trouver les solutions d'une requête SPARQL Q avec un ensemble de contraintes c. Les expressions équivalentes de celles-ci sont décrites :

+ Les variables dans $\mathcal{V}_U$ et $\mathcal{V}_I$ sont utilisées comme variables principales dans la requête SPARQL Q sur les graphes de connaissances RDF associés à G_U et G_I.
+ Les contraintes $c \in \mathcal{C}_F$ doivent être satisfaites en incorporant la clause FILTER dans la requête SPARQL Q.

```
1  PREFIX uvso: <http://utc.fr/uvso/ns#>
2  PREFIX uvo: <http://utc.fr/uvo/ns#>
3  PREFIX uvoo: <http://utc.fr/uvoo/ns#>
4  PREFIX rdf: <http://w3.org/1999/02/22-rdf-syntax-ns#>
5  PREFIX xsd: <http://w3.org/2001/XMLSchema#>
6  PREFIX gr: <http://purl.org/goodrelations/v1#>
7
8  SELECT ?auto
9  WHERE {
10         ?auto rdf:type uvso:Automobile.
11         ?auto uvso:couleur ?couleur.
12         FILTER contains(?couleur, "noir").
13         ?auto uvso:nombreDePlaces ?places.
14         ?places gr:aValeurEntier "5"^^xsd:int.
15         ?auto uvso:AFabricant ?marque.
16         FILTER (contains(str(?marque), "audi")).
17         ?auto uvso:StyleVehicule uvso:berline_occasion.
18         ?auto uvso:KiloetrageOdometre ?kilometrage.
19         ?kilometrage gr:aValeurFloat ?valeurKilometrage.
20         FILTER (?valeurKilometrage <= 100000) .
21         ?auto uvo:Estimation ?estimation.
22         ?estimation uvoo:aValeurMonetaire ?prix.
23         FILTER (?prix <= 100000 && ?prix >= 20000) .
24 } LIMIT 10
```

FIGURE 1 – Une requête SPARQL en correspondance avec des préférences de l'utilisateur.

La mise en correspondance de motifs de graphes est essentiellement le mécanisme utilisé par SPARQL pour récupérer des informations à partir de graphes de connaissances RDF. Dans ce contexte, une contrainte est considérée comme une évaluation d'un motif de graphe sur le graphe de connaissances RDF. Pour trouver des solutions, les requêtes SPARQL peuvent utiliser des motifs de triplets et des modificateurs de solutions en tant que contraintes. Les motifs de triplets impliquent trois variables et les modificateurs de solutions tels que ORDER BY, DISTINCT et LIMIT peuvent être utilisés pour trier, éliminer les doublons et limiter des solutions. Cette approche bénéficie de l'expressivité des requêtes SPARQL, qui ont le pouvoir expressif de l'algèbre relationnelle.

4 Expérimentations

Afin d'évaluer l'approche proposée, nous utilisons le graphe de connaissances RDF composé de 5537 individus de descriptions de véhicules et de 367 préférences d'utilisateurs, qui contiennent un total de 822 000 triplets RDF basés sur les modèles de l'ontologie présentés dans [13]. À partir d'une étude empirique des ensembles de données, nous montrons comment fonctionne notre SdR basé sur

des contraintes via le graphe de connaissances RDF. Tout d'abord, nous organisons les préférences des utilisateurs et les descriptions de véhicules en triplets RDF basés sur le modèle de l'ontologie afin de collecter les données de manière formelle. La construction d'un SdR basé sur des contraintes se concentre ensuite sur la résolution de deux ensembles de contraintes : des contraintes de connaissances spécifiques au domaine et des contraintes de préférences utilisateur. En particulier, l'ensemble de contraintes basé sur les connaissances spécifiques au domaine est traduit en règles SWRL et directement implémenté sur le graphe de connaissances RDF via des modules de raisonnement. La déduction d'informations nouvelles et pertinentes sur chaque utilisateur et chaque élément de véhicule est ensuite ajoutée à l'ensemble de données comme illustré dans la figure 2. L'ensemble de contraintes repose sur les préférences des utilisateurs liées aux informations sur leurs véhicules préférés, qui jouent un rôle essentiel dans la recherche de recommandations pertinentes. Par conséquent, nous formulons ces contraintes à l'aide de requêtes SPARQL basées sur la mise en correspondance de motifs sur des graphes et des modificateurs de solutions, comme indiqué dans la Figure 1. Dans le cas idéal, toutes les variables peuvent être affectées et nous pouvons trouver des solutions vérifiant pour le graphe de connaissances RDF. Au final, les recommandations produites respectent l'ensemble des contraintes et sont donc les plus pertinentes.

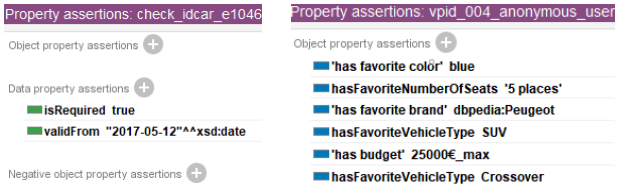


FIGURE 2 – Information déduite en utilisant les règles SWRL traduites à partir des contraintes $\mathcal{C}_{KB1}$ et $\mathcal{C}_{KB2}$.

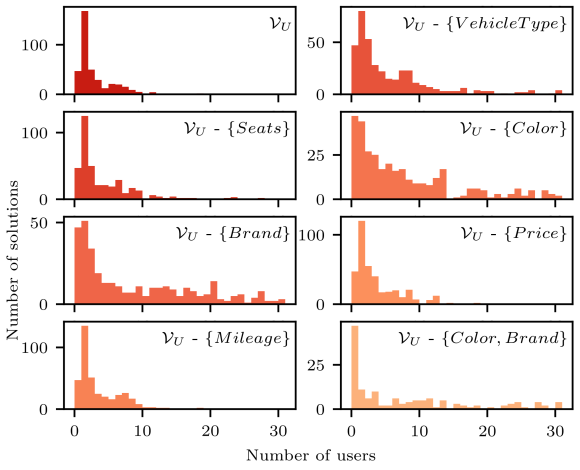


FIGURE 3 – Histogrammes représentant la distribution des solutions à travers différents ensembles de contraintes.

Cependant, de nombreux cas peuvent ne pas trouver de solution en raison de certaines incohérences entre les contraintes des préférences de l'utilisateur et les descriptions des véhicules. Il y a deux propositions possibles pour ce problème : (1) Enrichir et agrandir la base de connaissances RDF en augmentant le nombre de véhicules échangés sur les portails ; (2) Traiter et identifier un ensemble minimal de contraintes à partir des préférences de l'utilisateur. Au lieu d'enrichir le jeu de données comme dans la première proposition, la seconde proposition repose sur l'élimination ou l'adaptation des contraintes de l'utilisateur en utilisant un ensemble de diagnostic qui est défini comme un ensemble de contraintes Δ extrait de l'ensemble de contraintes $\mathcal{C}_F$ tel que les recommandations à partir du nouvel ensemble de contraintes $\mathcal{C}_F - \Delta$ sont cohérentes.

Afin d'expérimenter ce type de situation, nous avons construit des contraintes de préférences utilisateur basées sur l'ensemble de variables $\mathcal{V}_U = \{Seats, VehicleType, Brand, Color, Mileage, Price\}$ extraites des préférences utilisateur dans l'ensemble de données. L'expérience avec toutes les contraintes basées sur les préférences de l'utilisateur a abouti à 88 % des utilisateurs qui ont trouvé au moins une solution dans l'ensemble de données RDF. Avec la deuxième expérience, nous avons cherché à construire des ensembles de diagnostics afin de maximiser le nombre de solutions correspondant aux préférences de l'utilisateur et de réduire le nombre d'utilisateurs qui ne peuvent pas trouver de solution. Les ensembles de diagnostics incluent uniquement les contraintes éliminées de chaque préférence de l'utilisateur, en fonction d'un ordre de préférence défini par l'utilisateur pour ses préférences. Par exemple, $\Delta_1 = \{Places - Seats\}$, $\Delta_2 = \{TypeVehicule - VehicleType\}$, $\Delta_3 = \{Marque - Brand\}$, $\Delta_4 = \{Couleur - Color\}$, $\Delta_5 = \{Kilométrage - Mileage\}$, $\Delta_6 = \{Prix - Price\}$, $\Delta_7 = \{Couleur, Marque\}$. La figure 3 montre des histogrammes sur la distribution du nombre de solutions sur le nombre d'utilisateurs en utilisant différents ensembles de diagnostics. Avec toutes les contraintes $\mathcal{V}_U$, la majorité du nombre de solutions se situe dans une plage de 0 à 5 solutions par utilisateur. En appliquant des ensembles de diagnostics, le nombre de solutions s'étend avec une augmentation du nombre de solutions supérieure à 10 pour les utilisateurs. Ces changements dans le nombre de solutions sont particulièrement illustrés par l'ensemble de contraintes : $\mathcal{V}_U - \Delta_3$ et $\mathcal{V}_U - \Delta_4$. Pour réduire le nombre d'utilisateurs qui ne peuvent pas trouver de solution, l'élimination de plusieurs préférences de l'utilisateur peut devenir nécessaire. Cela signifie qu'il est nécessaire de faire un compromis entre la satisfaction de l'utilisateur et les résultats de recommandations.

Nous illustrons le SdR basé sur des contraintes à partir d'un graphe de connaissances RDF dans le domaine des véhicules. Les expérimentations menées confirment l'intérêt de notre approche de séparation des ensembles de contraintes en ensembles de contraintes de connaissances spécifiques au domaine et en ensembles de contraintes de préférences de l'utilisateur. En utilisant des règles SWRL, l'ensemble de connaissances spécifiques au domaine peut être déduit et intégré dans l'ensemble de données RDF. L'ensemble de contraintes construit sur les préférences utilisateur est traduit en requêtes SPARQL. Les recommandations perti-

nentes pour les utilisateurs sont extraites à partir des solutions obtenues par la recherche de motifs sur les graphes RDF.

5 Conclusion et perspectives

Dans cet article, nous avons présenté comment construire un SdR basé sur des contraintes s'appuyant sur un graphe de connaissances. Un tel système permet d'intégrer dans un modèle sémantique uniforme la description des éléments et celle du domaine dans lequel ils évoluent. Nous avons montré comment distinguer contraintes selon qu'elles concernent les connaissances spécifiques au domaine ou les préférences de l'utilisateur. En utilisant des règles SWRL, nous avons traduit les contraintes de connaissances spécifiques au domaine en règles et avons effectué des déductions de nouvelles informations pertinentes sur le graphe de connaissances RDF. Les contraintes de préférences de l'utilisateur peuvent être directement traduites en requêtes SPARQL. Nous avons mené une expérience sur notre approche basée sur le graphe de connaissances RDF de l'achat et de la vente de véhicules. Les résultats de recommandation obtenus à partir du SdR basé sur des contraintes sont prometteurs. Dans nos travaux futurs, nous prévoyons de rechercher l'exploitation d'ensembles de diagnostics, qui devraient être optimisés pour chaque utilisateur et pourraient aider à réduire le temps de performance pour proposer des recommandations pertinentes.

Remerciements

Cette recherche a été financée par l'Agence National de la Recherche (ANR) et par l'entreprise Vivocaz au titre du projet France Relance – préservation de l'emploi R&D (ANR-21-PRRD-0072-01).

Références

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6) :734–749, 2005.

[2] Muesluem Atas, TN Trang Tran, Alexander Felfernig, Seda Polat Erdeniz, Ralph Samer, and Martin Stettinger. Towards similarity-aware constraint-based recommendation. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 287–299. Springer, 2019.

[3] Boudjemaa Boudaa, Djamila Figuir, Slimane Hammoudi, and Sidi mohamed Benslimane. Datatourist : A constraint-based recommender system using datatourisme ontology. *International Journal of Decision Support System Technology*, 13(2) :62–84, 2021.

[4] Derek Bridge, Mehmet H Göker, Lorraine McGinty, and Barry Smyth. Case-based recommender systems. *The Knowledge Engineering Review*, 20(3), 2005.

[5] Camélia Dadouchi, Bruno Agard, and Benoit Montreuil. Context-aware interactive knowledge-based recommendation. *SN Computer Science*, 3(6), 2022.

[6] Alexander Felfernig. Application of constraint-based technologies in financial services recommendation. In *CEUR Workshop*, 2016.

[7] Alexander Felfernig and Robin Burke. Constraint-based recommender systems : technologies and research issues. In *Proceedings of the 10th international conference on Electronic commerce*, 2008.

[8] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 11(2) :11–34, 2006.

[9] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. Developing constraint-based recommenders. In *Recommender systems handbook*, pages 187–215. Springer, 2011.

[10] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. Constraint-based recommender systems. In *Recommender systems handbook*, pages 161–190. Springer, 2015.

[11] Dietmar Jannach, Markus Zanker, and Matthias Fuchs. Constraint-based recommendation in tourism : A multiperspective case study. *Information Technology & Tourism*, 11(2) :139–155, 2009.

[12] Luyen Le Ngoc, Marie-Hélène Abel, and Philippe Gouspillou. Apport des ontologies pour le calcul de la similarité sémantique au sein d'un système de recommandation. In *Ingénierie des Connaissances (Evènement affilié à PFIA Plate-Forme Intelligence Artificielle)*, 2022.

[13] Luyen Le Ngoc, Marie-Hélène Abel, and Philippe Gouspillou. Towards an ontology-based recommender system for the vehicle sales area. In *International Conference on Deep Learning, Artificial Intelligence and Robotics*, pages 126–136. Springer, 2022.

[14] Luyen Le Ngoc, Jinfeng Zhong, Elsa Negre, and Marie-Hélène Abel. Constraint-based recommender system for crisis management simulations. In *The 56th Hawaii International Conference on System Sciences*, 2023.

[15] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. Recommender system application developments : a survey. *Decision Support Systems*, 74 :12–32, 2015.

[16] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10) :2004, 2004.

[17] Maryam Ramezani, Lawrence Bergman, Rich Thompson, Robin Burke, and Bamshad Mobasher. Selecting and applying recommendation technology. In *International Workshop on Recommendation and Collaboration in Conjunction with 2008 International ACM Conference on Intelligent User Interfaces, IUI*, pages 613–620, 2008.

Un cadre pour inclure et exploiter des informations probabilistes dans les rapports de validation SHACL

Rémi Felin[1], Catherine Faron[1], Andrea G. B. Tettamanzi[1]

[1] Université Côte d'Azur, Inria, I3S, Sophia-Antipolis, France

Résumé

SHACL est une recommandation du W3C qui permet de représenter en RDF des contraintes appelées formes (shapes en anglais) et de valider des graphes de données RDF par rapport à ces contraintes. Un validateur SHACL produit un résultat booléen, faux pour une forme SHACL quand au moins un triple dans le graphe RDF n'est pas conforme à la forme, vrai sinon. Nous proposons un cadre probabiliste pour valider un graphe RDF avec une proportion réaliste de triplets qui ne se conforment pas à une shape.

Mots-clés

RDF, SHACL, Validation de données, Evaluation probabiliste

Abstract

SHACL is a W3C recommendation to represent constraints in RDF– shape graphs –, and validate RDF data against these constraints. A SHACL validator outputs boolean results, false for a shape as soon as there is at least one triple in the RDF data that does not conform to the shape, else true. In this paper, we propose a probabilistic framework to validate an RDF graph with a realistic proportion of triples that does not conform to a shape.

Keywords

RDF, SHACL, Data Validation, Probabilistic Assessment

1 Introduction

Le développement du Web sémantique a conduit à l'émergence de nouveaux domaines de recherche tels que la qualité des données RDF. SHACL est le langage recommandé par le W3C pour représenter des contraintes que les données RDF doivent respecter afin d'assurer la cohérence d'un jeu de données. Un validateur SHACL produit un résultat booléen, faux pour une *shape* quand au moins un noeud dans le graphe RDF n'est pas conforme à la *shape*, vrai sinon. En considérant par exemple un grand ensemble de données RDF construit de manière collaborative avec une augmentation massive et constante de triples RDF (par exemple, DBpedia), la violation de contraintes SHACL semble inévitable en raison de données incomplètes et/ou incorrectes. Dans la pratique, un examen plus approfondi des données semble nécessaire. Un expert pourrait élaborer une stratégie de mise à jour des données ou des *shapes* en fonction du taux et/ou de la nature des violations. Nous abordons la question de recherche suivante :

> *Comment concevoir un processus de validation prenant en compte les erreurs physiologiques dans les données de la vie réelle ?*

Notre contribution aborde le problème en suggérant un cadre basé sur un modèle probabiliste afin de considérer un taux de violations de contraintes autorisé p égal à la proportion d'erreurs que les données RDF contiennent. Nous définissons une *mesure de la probabilité* d'observer un nombre donné de violations. Nous évaluons un graphe RDF par rapport à un ensemble de *shapes* en tenant compte d'un taux d'erreur physiologique théorique.

Le présent document est organisé comme suit : Dans la section 2, nous résumons les travaux connexes et le positionnement de notre travail. Dans la section 3, nous présentons notre modèle probabiliste (3.1), notre extension du modèle de rapport de validation SHACL (3.2) et notre proposition d'un processus de validation SHACL étendu (3.3). Nous présentons les résultats de nos expériences dans la section 4. Nous concluons et discutons des recherches futures dans la section 5.

2 Travaux connexes

SHACL [14] étant une recommandation assez récente (2017), ses relations avec d'autres standards font l'objet de recherches en cours. En particulier, nous trouvons des travaux sur ses relations avec les règles d'inférence [21], avec OWL [2], le raisonnement en logiques de descripton [16] et les patrons de conception d'ontologies [19]. De plus, des extensions concernant la validation SHACL émergent, par exemple un moteur de validation SHACL basé sur l'étude de la connectivité d'un graphe RDF et la collecte de données dans ce même graphe [12]. L'expressivité et la sémantique de SHACL sont un sujet riche dans la littérature [1, 16] : ces travaux ont mis en évidence une sémantique basée sur $\mathcal{SROIQ}$, l'une des logiques de description les plus expressives.

La validation de données RDF avec SHACL est une question de recherche largement abordée dans la littérature [3, 8, 10, 13, 15, 20]. Tous ces travaux considèrent une utilisation standard de SHACL : un graphe RDF est valide par rapport à une *shape* s'il vérifie les contraintes exprimées. Notre approche étend le processus de validation SHACL standard

pour dépasser son caractère binaire en considérant un taux de violation de contrainte acceptable.

D'autres travaux s'intéressent à la génération de contraintes SHACL [11, 23, 24]. Différentes approches conduisent à différentes façons de traiter la validation de ces *shapes*. Certaines approches exploitent des données RDF et des statistiques, et nécessitent une analyse d'expert pour valider une *shape* candidate. Le profilage des graphes de connaissances [22] est une approche possible pour induire des contraintes à partir de grands graphes RDF. Une de ces approches [18] s'appuie sur des techniques d'apprentissage automatique pour générer automatiquement des *shapes* en utilisant des données RDF profilées comme caractéristiques. Certaines approches exploitent des ontologies pour générer des *shapes* [5] : notamment les signatures de propriétés ; dans ce cas les *shapes* générées peuvent être considérées valides si l'ontologie est de bonne qualité.

Le travail présenté dans cet article est axé sur la validation des données RDF par rapport à des *shapes* et vise à fournir une expertise sur la cohérence des données RDF par rapport à un ensemble de *shapes* (qui peuvent avoir été générées automatiquement ou fournies par un expert), et en acceptant un taux d'erreurs physiologique que les données peuvent contenir.

3 Un cadre probabiliste pour l'évaluation de *shapes* SHACL

3.1 Modèle probabiliste

Dans un contexte réel, les ensembles de données RDF sont imparfaits, incomplets (dans le sens où des données attendues sont manquantes) et contiennent des erreurs de différentes natures. Le contrôle de la qualité des données RDF et l'intégration efficace des données, garantissant la cohérence des données RDF, sont des cas d'utilisation qui peuvent être traités à l'aide de SHACL. Par ailleurs, l'extraction de *shapes* SHACL à partir de données RDF est une approche prometteuse pour apprendre la connaissance du domaine (contraintes du domaine). Les *shapes* candidates sont celles qui déclenchent quelques violations dans les données, mais cela est directement corrélé à la qualité (taux d'erreur, qui est cependant inconnu) de l'ensemble de données RDF considéré.

Nous proposons d'étendre l'évaluation des données RDF par rapport aux *shapes* en considérant une proportion d'erreur théorique physiologique p dans les données RDF réelles. Dans ce contexte, la modélisation mathématique du processus d'évaluation SHACL qui tient compte d'une proportion d'erreur p est basée sur un modèle probabiliste.

La cardinalité de référence. (ou cardinalité du support) d'une *shape* S, υ_s, est l'ensemble des triplets RDF concernés par S et testés durant la validation. On la note $\|\upsilon_s\|$.

Les confirmations et les violations. On note υ_s^+ et υ_s^- les ensembles disjoints, respectivement, des triplets qui sont conformes à S et des triplets qui violent S. Le support d'une *shape* S est l'union disjointe de ses confirmations et violations :

$$\upsilon_S = \upsilon_S^+ \cup \upsilon_S^- \tag{1}$$

Modélisation du processus de validation. Soit X une variable aléatoire qui conceptualise un ensemble d'observations provenant de la validation d'une *shape* S, c'est-à-dire un ensemble de triplets RDF υ_S où chaque triplet $t \in \upsilon_S$ peut être soit une *confirmation* ($t \in \upsilon_S^+$) soit une *violation* ($t \in \upsilon_S^-$)

Soit un triplet t, tiré au hasard dans υ_S ; nous pouvons définir, à partir de ce triplet, une variable aléatoire qui prend deux valeurs : **1** si $t \in \upsilon_S^-$ et **0** sinon. Nous en concluons qu'une loi binomiale peut modéliser cette approche probabiliste : $B(\|\upsilon_S\|, p)$ avec p la proportion d'erreur théorique.

Mesure de vraisemblance. On note L_k la vraisemblance d'obtenir k violations ($\|\upsilon_S^-\| = k$) parmi n triplets ciblés ($n = \|\upsilon_S\|$) : $L_k = P(X = k)$. En considérant que X suit la loi binomiale $B(\|\upsilon_S\|, p)$, on a :

$$L_{\|\upsilon_S^-\|} = P(X = \|\upsilon_S^-\|) = \binom{\|\upsilon_S\|}{\|\upsilon_S^-\|} \cdot p^{\|\upsilon_S^-\|} \cdot (1-p)^{\|\upsilon_S^+\|}. \tag{2}$$

3.2 Extension du modèle du rapport de validation SHACL

Nous proposons un modèle enrichi du rapport de validation SHACL afin d'exprimer des informations supplémentaires pour chaque *shape* considérée dans le rapport. Nous avons défini une extension du vocabulaire du rapport de validation SHACL, dans un espace de noms dénotée par le préfixe `psh` dans la suite. [1]. Pour chaque *shape* source considérée dans la validation d'un graphe RDF, nous générons des triplets supplémentaires : la propriété `psh:summary` relie le rapport de validation à un nœud blanc de type `psh:ValidationSummary`, qui est le sujet de plusieurs propriétés dont les valeurs sont le résultat du calcul de différentes mesures relatives à la *shape* source.

La *shape* ciblée. Il s'agit de la valeur de la propriété `psh:focusShape`. C'est la source de la *shape* de validation qui est ensuite décrite dans le résumé de validation.

La cardinalité de référence. $\|\upsilon_S\|$, est la valeur de la propriété `psh:referenceCardinality`.

Le nombre de confirmations et de violations. Respectivement $\|\upsilon_S^+\|$ et $\|\upsilon_S^-\|$, sont les valeurs des propriétés `psh:numConfirmation` et `psh:numViolation`.

La généralité $G(S) \in [0, 1]$ mesure la *représentativité* de S en considérant l'ensemble du graphe RDF υ :

$$G(S) = \frac{\|\upsilon_S\|}{\|\upsilon\|} \tag{3}$$

C'est la valeur de la propriété `psh:generality`.

La vraisemblance $L_{\|\upsilon_S^-\|}$ d'une *shape* S dans un graphe RDF υ telle que définie dans la Section 3.1 est la valeur de la propriété `psh:likelihood`.

La figure 1 présente un extrait d'un exemple de rapport de validation dans lequel :

1. `prefix psh: <http://ns.inria.fr/probabilistic-shacl/>`

```
[ a sh:ValidationReport ;
    sh:conforms boolean ;
    sh:result r ;
    # Probabilistic SHACL extension
    psh:summary [
        a psh:ValidationSummary ;
        psh:referenceCardinality ‖v_S‖ ;
        psh:numConfirmation ‖v_S^+‖ ;
        psh:numViolation ‖v_S^-‖ ;
        psh:generality G(S) ;
        psh:likelihood L_{‖v_S^-‖} ;
        psh:focusShape S
    ] ;
] .
```

FIGURE 1 – Structure du rapport de validation SHACL étendu

— l'URI `:s1` dénote une *shape* SHACL s_1 ;
— la cardinalité du graphe RDF en cours de validation est $\|v\| = 1000$;
— le paramètre de la distribution binomiale est $p = 0.1$.

3.3 Validation d'un graphe RDF par rapport à une *shape* SHACL comme un test d'hypothèse

Le processus de validation d'un graphe RDF par rapport à une *shape* donnée S est basé sur le modèle probabiliste proposé dans la section 3.1, qui repose sur l'hypothèse selon laquelle une observation donnée suit une distribution binomiale $X \sim B(\|v_S\|, p)$. Pour valider cette hypothèse, nous procédons à un test d'hypothèse.
La validation d'un graphe RDF par rapport à une *shape* S est basée sur la proportion observée de violations de S dans le graphe, notée $\hat{p} : \hat{p} = \frac{\|v_S^-\|}{\|v_S\|}$. Un graphe RDF est valide par rapport à S si la proportion observée de violation de S est inférieure à la proportion théorique :

$$\hat{p} \leq p \implies v \models S \tag{4}$$

Dans le cas où la proportion observée est supérieure à la proportion théorique, nous évaluons la distance de cette probabilité à partir des valeurs maximales de la fonction de masse de la distribution binomiale $B(\|v_S\|, p)$ en utilisant les tests d'hypothèses. La figure 3 montre la proportion du nombre de violations que nous acceptons par rapport au nombre que nous rejetons avec notre méthode.

L'hypothèse nulle (H_0) et l'hypothèse alternative (H_1). L'hypothèse nulle est que *les données suivent la distribution donnée*, c'est-à-dire que la fréquence des violations observées $\hat{p} = \frac{\|v_S^-\|}{\|v_S\|}$ est conforme aux proportions attendues de violations p et $X \sim B(\|v_S\|, p)$. L'hypothèse alternative H_1 est que *les données ne suivent pas la distribution donnée*.

Le test d'ajustement. Ce test d'hypothèse vérifie l'alignement de nos observations avec une distribution théorique : nous définissons X_s^2 la **statistique de test** pour une *shape* S qui suit $\chi^2_{k-1,\alpha}$ en supposant H_0, c'est-à-dire que $X_s^2 \sim \chi^2_{k-1,\alpha}$ (une distribution du chi-carré avec $k-1$ degrés de liberté et un seuil de signification de $1-\alpha$). Ce test est effectué au seuil α défini à 5%. Il considère k comme le nombre total de groupes, c'est-à-dire $k = 2$, n_i le nombre d'individus observés et T_i le nombre d'individus théoriques. La statistique de test X_s^2 est définie par

$$X_s^2 = \sum_{i=1}^{k} \frac{(n_i - T_i)^2}{T_i} \sim \chi^2_{k-1;\alpha}. \tag{5}$$

.
Le test d'ajustement (Formule 5) est applicable si $\forall i \in [1, k], T_i \geq 5$. Supposons une *shape* S pour laquelle nous observons un très faible support $\|v_S\|$ (supposons $\|v_S\| = 8$) implique une proportion de violations et/ou de confirmations inférieure à 5. Dans ce cas, le test d'hypothèse ne peut pas être réalisé car l'échantillon n'est pas suffisamment représentatif.

La valeur critique. La valeur à partir de laquelle on rejette l'hypothèse nulle H_0, est égale à $\chi^2_{k-1;\alpha}$. En prenant $\alpha = 0.05$ et $k = 2$, on a $\chi^2_{k-1;\alpha} = \chi^2_{1;\alpha=0.05} = 3.84$.
Une formule alternative considère l'intervalle d'acceptation I_a d'une distribution du chi-carré, c'est-à-dire $I_a = [0, \chi^2_{k-1;\alpha}]$ qui accepte H_0 si $X_S^2 \in I_a$.

L'acceptation de l'hypothèse nulle. Accepter H_0 et donc $X \sim B(\|v_S\|, p)$, implique que la valeur de notre statistique de test X_S^2 n'est pas incluse dans la zone de rejet de la distribution $\chi^2_{k=1}$:

$$X_S^2 \leq \chi^2_{k-1;\alpha}. \tag{6}$$

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix psh:
      <http://ns.inria.fr/probabilistic-shacl/> .
@prefix : <http://www.example.com/myDataGraph#> .

# SHACL Standard
:v1 a sh:ValidationResult ;
    sh:focusNode :n1 ;
    [...]
    sh:sourceShape :s1 .

:v2 a sh:ValidationResult ;
    sh:focusNode :n2 ;
    [...]
    sh:sourceShape :s1 .

[...]

[ a sh:ValidationReport ;
    sh:conforms false ;
    sh:result :v1 ;
    sh:result :v2 ;
    [...]
    # SHACL Extension
    # shape s1
    psh:summary [
        a psh:ValidationSummary ;
        psh:generality "0.2"^^xsd:decimal ;
        psh:numConfirmation 178 ;
        psh:numViolation 22 ;
        psh:likelihood "0.0806"^^xsd:decimal ;
        psh:referenceCardinality 200 ;
        psh:focusShape :s1
    ] ;
] .
```

FIGURE 2 – Exemple d'un rapport de validation SHACL étendu pour une *shape* `:s1`, calculé avec $\|v\| = 1000$ et $p = 0.1$

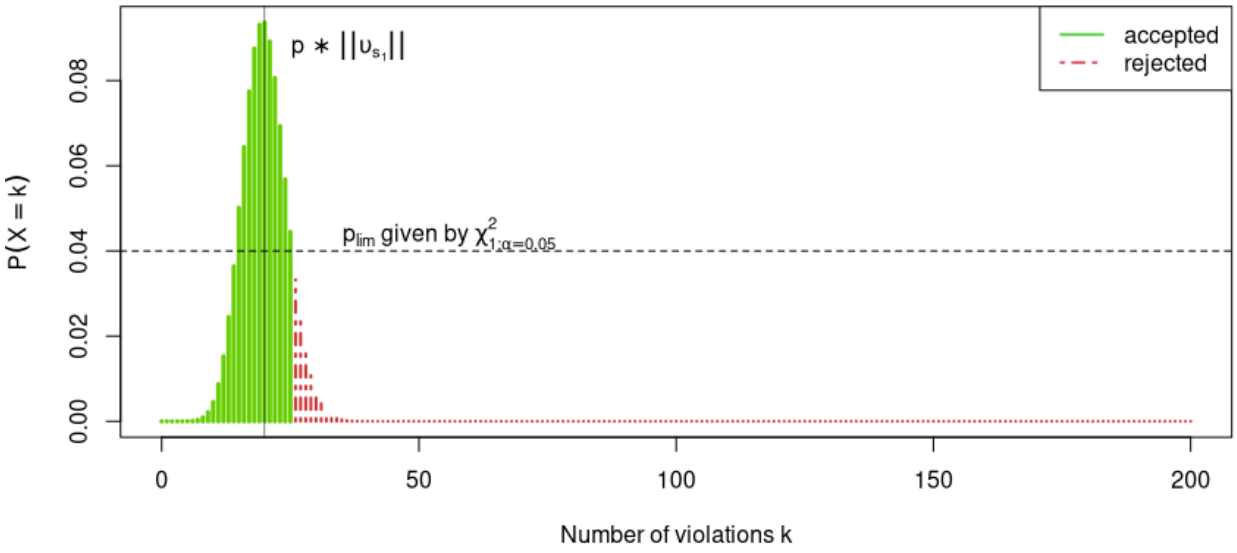


FIGURE 3 – Zone d'acceptation d'une *shape* s_1 considérant $X \sim B(\|\upsilon_{s_1}\|, p)$ où $\|\upsilon_{s_1}\| = 200$ et $p = 0.1$.

L'acceptation de H_0 implique la validation du graphe RDF par rapport à la *shape* S considérée, c'est-à-dire,

$$X_S^2 \leq \chi^2_{k-1;\alpha} \implies \upsilon \models S. \tag{7}$$

Prenons l'exemple de la Figure 2. Nous observons une proportion de violations légèrement supérieure à celle attendue, c'est-à-dire, $\hat{p} = \frac{\|\upsilon_{s_1}^-\|}{\|\upsilon_{s_1}\|} = 0.11$ et $\hat{p} > p$: le test d'hypothèse permet de déterminer si cette observation est compatible ou non avec l'hypothèse nulle, et dans cet exemple, nous rejetterions H_0 et ne validerions pas le graphe par rapport à la *shape* `:s1`. En prenant $\alpha = 5\%$ pour évaluer $X^2_{s_1}$, on obtient :

$$X^2_{s_1} = \frac{(22-20)^2}{20} + \frac{(178-180)^2}{180} \approx 0.222.$$

Ainsi, le test statistique a montré que $X^2_{s_1} \leq \chi^2_{1;\alpha=0.05}$ (c'est-à-dire 3.84) et donc $X^2_{s_1} \in I_a$. Nous acceptons H_0 i.e. l'hypothèse que nos observations sur la conformité des triplets à `:s1` suivent une distribution binomiale $X \sim B(200, 0.1)$.

4 Expériences

Nous avons implémenté le modèle proposé dans un moteur de validation probabiliste SHACL reposant sur le moteur sémantique *Corese* [6]. Le rapport de validation étendu fournit un degré de probabilité exprimé sous l'hypothèse que les échantillons suivent une distribution binomiale avec une cardinalité définie pour une *shape* S (c'est-à-dire $\|\upsilon_S\|$) et une probabilité p définie *empiriquement* correspondant à la proportion supposée de violations que nous acceptons pour certaines données RDF. En considérant un ensemble de contraintes SHACL représentatives d'un (large) graphe RDF donné, la recherche d'un taux d'erreur p pour lequel il est raisonnable de considérer l'acceptation de données est une manière d'évaluer ce travail. Cela implique une analyse détaillée des caractéristiques du graphe RDF considéré, des proportions de *shapes* acceptées ou rejetées et de l'impact des tests d'hypothèse sur l'acceptation.

4.1 Protocole expérimental

Nos expériences utilisent le jeu de données RDF *CovidOnTheWeb* [2] [17] et un ensemble de 377 *shapes* SHACL construites à partir de règles d'association issues de la fouille de *CovidOnTheWeb* [4] et considérées comme représentatives de ce graphe.
Nous effectuons une analyse du taux d'erreur théorique afin de trouver empiriquement un taux optimal : nous testons les 20 valeurs de $p \in \{0,05, 0,1, 0,15, \ldots, 0,95, 1\}$. Les expériences ont été effectuées sur un Dell Precision 3561 équipé d'un processeur Intel(R) Core i7-11850H de 11e génération, avec 32 Go de RAM fonctionnant sous le système d'exploitation Fedora Linux 35. Le code source est disponible dans un dépôt public. [3]

CovidOnTheWeb. Il s'agit d'un graphe de connaissances RDF produit à partir du *COVID-19 Open Research Dataset (CORD-19)*. Il décrit des articles scientifiques, identifiés par des URI et associés aux entités nommées extraites dans ces articles, désambiguïsées par *Entity-Fishing* et liées à des entités *Wikidata*. La figure 4 montre un extrait de description RDF dans CovidOnTheWeb au format *turtle* et le tableau 1 montre les caractéristiques du jeu de données RDF. Nous

2. https ://github.com/Wimmics/CovidOnTheWeb
3. https://github.com/RemiFELIN/RDFMining/tree/eswc_2023

TABLE 1 – Résumé du sous-graphe de données RDF *CovidOnTheWeb* considéré pour les expériences.

#triplets RDF	226,647
#articles distincts	20,912
#entités nommées distinctes	6,331
moyenne #entités nommées par article	10.52

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax
    -ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    .
@prefix covid:    <http://ns.inria.fr/covid19/> .
@prefix entity:   <http://www.wikidata.org/entity/> .

covid:ec1[...]2c5   rdf:type    entity:Q4407  .
covid:fff[...]86d   rdf:type    entity:Q10876 .
[...]
entity:Q4407    rdfs:label  "methyl"@en .
entity:Q10876   rdfs:label  "bacteria"@en .
```

FIGURE 4 – Exemple d'un extrait de données RDF du sous-graphe *CovidOnTheWeb*

considérons un sous-ensemble contenant environ 18,79% des articles et 0,01% des entités nommées.

Les *shapes* candidates. Ces *shapes* représentent les règles d'association obtenues par Cadorel et al.[4] à partir d'un sous-ensemble du jeu de données *CovidOnTheWeb*. Ces règles ne sont pas nécessairement parfaites, nous nous intéressons donc à les utiliser dans notre approche probabiliste. À partir des résultats expérimentaux de Cadorel et al., nous avons extrait les entités nommées correspondant aux antécédents et conséquents de ces règles d'association. Nous avons effectué un traitement permettant la conversion de ces règles en *shapes* SHACL. Nous ciblons les articles appartenant à une entité nommée, représentant l'*antécédent*, avec la propriété `sh:targetClass`. Parmi les articles considérés, nous cherchons à déterminer l'affiliation à une autre entité nommée, représentant le *conséquent* : nous utilisons une contrainte appliquée sur le type d'article et ciblant une entité nommée avec la propriété `sh:hasValue`. Dans ce contexte, une violation invoquera une violation de type `sh:HasValueConstraintComponent` pour la *shape* courante. Un exemple de *shape* formée après traitement est présenté dans la figure 5.

4.2 Résultats

Le tableau 2 présente les premiers résultats expérimentaux, notamment le score de généralité qui est relativement faible : la cardinalité de référence moyenne est assez faible par rapport au nombre total de triplets RDF dans notre ensemble de données : environ 106 triplets RDF en moyenne sont ciblés par nos *shapes* (0,047% des triplets RDF). Le taux de violations est relativement élevé mais cela est nuancé par le taux de confirmations (33,19%).

La figure 6a montre une évolution croissante de la mesure de vraisemblance jusqu'à la valeur $p = 0.5$ puis une diminution. Il apparaît ainsi que le taux d'erreur le plus raisonnable est 50%, car il maximise la valeur moyenne de

```
@prefix : <http://www.example.com/myDataGraph#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix entity: <http://www.wikidata.org/entity/> .

:1 a sh:NodeShape ;
    sh:targetClass  entity:Q10295810 ;
    sh:property [
        sh:path rdf:type ;
        sh:hasValue entity:Q43656 ;
    ] .
```

FIGURE 5 – Exemple d'une *shape* SHACL représentant une règle d'association : `entity:Q10295810` (`"hypocholesterolemia"@en`) en tant qu' *antecedent* et `entity:Q43656` (`"cholesterol"@en`) en tant que *consequent*.

TABLE 2 – Résumé de la validation du graphe de *shapes* SHACL.

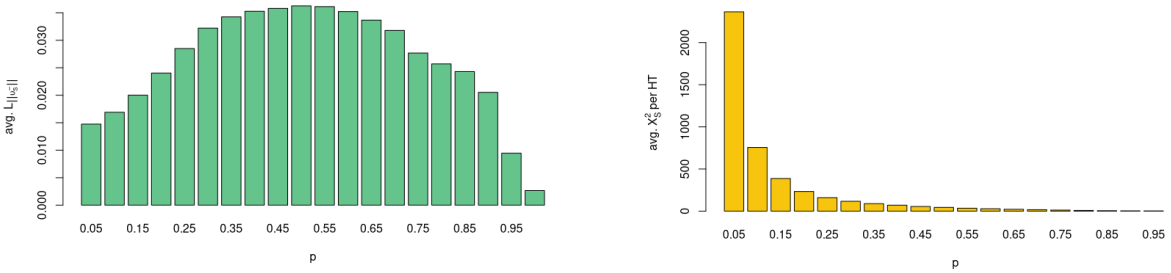
#entités nommées représentées	337 (*5.32%*)
moyenne $\|v_{\mathbf{S}}\|$	106.69 (*0.0470%*)
moyenne $\|v_{\mathbf{S}}^{+}\|$	33.19 (*31.11%*)
moyenne $\|v_{\mathbf{S}}^{-}\|$	73.50 (*70.89%*)
moyenne $\mathbf{G}(\mathbf{S})$ (Formule 3)	0.0005%

vraisemblance (0,0362%).

La figure 7 présente l'ensemble des décisions prises sur les *shapes* (acceptation, rejet) en fonction de la proportion théorique d'erreurs p et montre clairement l'importance des tests d'hypothèse. Le nombre de tests effectués augmente jusqu'à $p = 0.3$ puis diminue. De même, les tests d'hypothèse ont tendance à rejeter les *shapes* pour des valeurs "petites" de p et la tendance s'inverse à mesure que p augmente : le nombre de *shapes* acceptées augmente et la valeur de la statistique de test diminue (voir figure 6b). Une analyse plus poussée des résultats obtenus avec $p = 0.5$ montre que 63 *shapes* parmi les 187 *shapes* acceptées sont acceptées après avoir effectué un test d'hypothèse, c'est-à-dire 33,7% des *shapes* acceptées. Ces mêmes tests ont accepté 25,7% des *shapes* qui ont été testées, ce qui montre leur capacité à filtrer efficacement les *shapes* non valides avec un risque $\alpha = 0,05$ (5%) d'être incorrect.

La production des résultats au format HTML a été effectuée avec une transformation STTL [7], une extension du langage de requête SPARQL pour transformer RDF en n'importe quel format textuel Un extrait des rapports obtenus pour 20 *shapes* avec une proportion d'erreur théorique $p = 0.5$ est présenté dans la figure 8.

Nous avons comparé le temps de calcul de notre cadre de validation probabiliste proposé à celui de la validation standard. Pour notre base de 377 *shapes* et notre extraction de *CovidOnTheWeb* (226,647 triplets), nous avons observé un temps de calcul global de 1 minute et 35 secondes pour le cadre de validation probabiliste contre 1 minute et 29 secondes pour la validation standard : le cadre probabiliste prend 6.31% de temps supplémentaire par rapport à la validation standard et il est linéaire, ce qui le rend pratique et évolutif.



(a) $L_{\|\upsilon_S^-\|}$ average

(b) X_S^2 average

FIGURE 6 – Valeur moyenne (a) des mesures de vraisemblance et (b) des tests statistiques, en fonction de la proportion d'erreur théorique p.

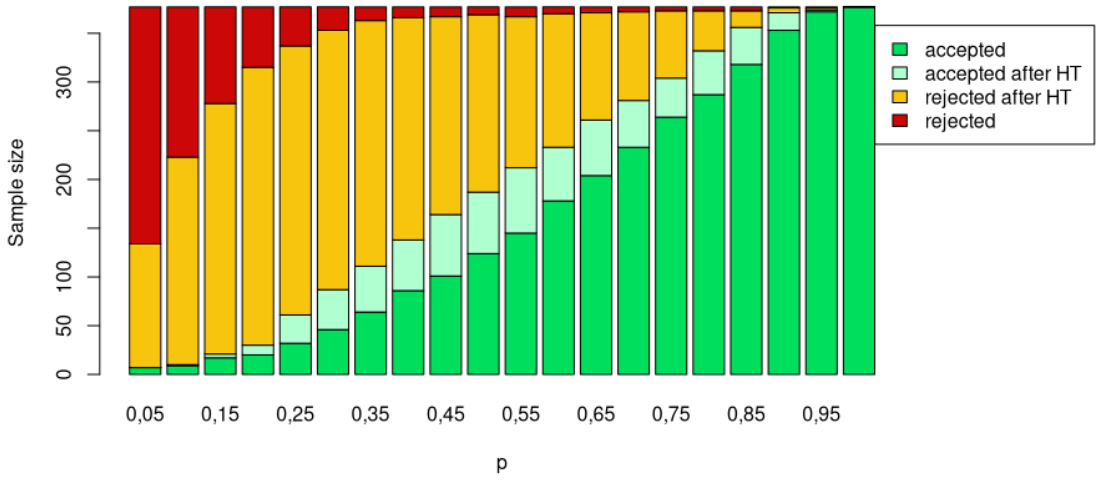


FIGURE 7 – Acceptation des *shapes* en fonction de la proportion d'erreur théorique p (HT = Test d'hypothèse)

5 Conclusion

Dans cet article, nous proposons un cadre probabiliste pour la validation SHACL, contribuant ainsi au contrôle de qualité des données RDF. Nous étendons le rapport de validation SHACL pour exprimer une mesure de vraisemblance pour le nombre de violations observées et proposons un modèle de décision pour une acceptation probabiliste des triplets RDF par rapport aux *shapes* SHACL. Nos expériences montrent les capacités de notre approche à valider un ensemble de *shapes* avec un taux d'erreur raisonnable p. Dans le cadre de travaux futurs, nous prévoyons d'étendre notre cadre proposé aux *shapes* complexes, en particulier les *shapes* récursives qui font l'objet de recherches en cours [3, 9, 20]. Nous prévoyons également d'étudier l'extraction automatique de *shapes* SHACL à partir de jeux de données RDF de référence, afin de capturer des connaissances de domaine sous forme de contraintes.

Remerciements

Ce travail a été partiellement financé par le projet 3IA Côte d'Azur "Investissements d'avenir" géré par l'Agence Nationale de la Recherche (ANR) avec le numéro de référence ANR-19-P3IA-0002.

Références

[1] Bart Bogaerts, Maxim Jakubowski, and Jan Van den Bussche. Expressiveness of shacl features. In *ICDT*, 2022.

[2] Bart Bogaerts, Maxime Jakubowski, and Jan Van den Bussche. Shacl : A description logic in disguise. 08 2021.

[3] Iovka Boneva, Jose G Labra Gayo, and Eric G Prud 'Hommeaux. Semantics and Validation of Shapes Schemas for RDF. In *ISWC2017 - 16th In-*

antecedent	consequent	referenceCardinality	#violation	likelihood	generality	X^2_S	Acceptance
two-hybrid screening	protein–protein interaction	48	19	0.041004880900459284	0.00021178308117910231		true
nidovirales	proteolysis	80	69	8.6669313322632E-12	0.00035297180196517053	42.05	false
intensive care medicine	acute respiratory distress syndrome	166	139	9.193409214822706E-20	0.0007324164890777288	75.56626506024097	false
astrocyte	central nervous system	70	34	0.09238587705330051	0.0003088503267195242		true
dopamine	serotonin	10	6	0.205078125	0.00004412147524564632	0.4	true
crystallography	crystal structure	20	7	0.0739288330078125	0.00008824295049129263		true
human parainfluenza	adenoviridae	237	133	0.00880821375320367	0.0010456789633218177	3.548523206751055	true
carbohydrate	lectin	114	75	2.4200572197826046E-4	0.000502984817800368	11.368421052631579	false
mycoplasma bovis	bovine coronavirus	12	6	0.2255859375	0.00005294577029477558		true
crystallization	diffraction	31	21	0.020653086248785257	0.00013677657326150358	3.903225806451613	false
membrane raft	methyl	32	19	0.08087921887636185	0.0001411887207860682	1.125	true
ifitm1	ifitm3	27	9	0.0349195740617752	0.00011912798316324504		true
multiple sclerosis	myelin	139	97	1.0209205741082355E-6	0.0006132885059144837	21.762589928057555	false
wheeze	asthma	85	44	0.08188889187584301	0.00037503253958799367	0.10588235294117647	true
influenza a virus subtype h5n1	avian influenza	277	165	2.969648471686876E-4	0.001222164864304403	10.140794223826715	false
hepatocellular carcinoma	liver cirrhosis	72	46	0.005843155895129734	0.00031767462176865343	5.555555555555555	false
diffraction	x-ray crystallography	16	7	0.174560546875	0.0000705943603930341		true
feline infectious peritonitis	feline coronavirus	130	46	2.605193913325792E-4	0.000573579178193402		true
aedes aegypti	culicidae	21	4	0.002853870391845703	0.00009265509801585726		true
monomer	oligomer	83	70	5.4692741602999564E-11	0.0003662082445388644	39.144578313253014	false

FIGURE 8 – Rapport de validation SHACL étendu en considérant $p = 0.5$.

ternational semantic web conference, Vienna, Austria, October 2017.

[4] Lucie Cadorel and Andrea Tettamanzi. Mining rdf data of covid-19 scientific literature for interesting association rules. *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 145–152, 2020.

[5] Andrea Cimmino, Alba Fernández-Izquierdo, and Raúl García-Castro. Astrea : Automatic generation of shacl shapes from ontologies. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web*, pages 497–513, Cham, 2020. Springer International Publishing.

[6] Olivier Corby, Rémi Ceres, Erwan Demairy, Fuqi Song, Virginie Bottollier, and Olivier Savoie. Corese : Semantic Web Factory. `https://project.inria.fr/corese/`.

[7] Olivier Corby and Catherine Faron Zucker. STTL : A SPARQL-based Transformation Language for RDF. In *11th International Conference on Web Information Systems and Technologies*, Lisbon, Portugal, May 2015.

[8] Julien Corman, Fernando Florenzano, Juan L. Reutter, and Ognjen Savkovic. Validating shacl constraints over a sparql endpoint. In *International Workshop on the Semantic Web*, 2019.

[9] Julien Corman, Juan L. Reutter, and Ognjen Savković. Semantics and validation of recursive shacl. In Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web – ISWC 2018*, pages 318–336, Cham, 2018. Springer International Publishing.

[10] Christophe Debruyne and Kris McGlinn. Reusable shacl constraint components for validating geospatial linked data (short paper). In *GeoLD@ESWC*, 2021.

[11] Daniel Fernandez-Álvarez, Jose Emilio Labra-Gayo, and Daniel Gayo-Avello. Automatic extraction of shapes using shexer. *Knowledge-Based Systems*, 238 :107975, 2022.

[12] Mónica Figuera, Philipp D. Rohde, and Maria-Esther Vidal. Trav-shacl : Efficiently validating networks of shacl constraints. In *Proceedings of the Web Conference 2021*, WWW '21, page 3337–3348, New York, NY, USA, 2021. Association for Computing Machinery.

[13] Ranjith K Soman. Modelling construction scheduling constraints using shapes constraint language (shacl). pages 351–358, 07 2019.

[14] Dimitris Kontokostas and Holger Knublauch. Shapes constraint language (SHACL). W3C recommendation, W3C, July 2017. https ://www.w3.org/TR/2017/REC-shacl-20170720/.

[15] Aljosha Köcher, Luis Miguel Vieira da Silva, and Alexander Fay. Constraint checking of skills using shacl. 07 2021.

[16] Martin Leinberger, Philipp Seifer, Tjitze Rienstra, Ralf Lämmel, and Steffen Staab. Deciding shacl shape containment through description logics reasoning. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 366–383, Cham, 2020. Springer International Publishing.

[17] Franck Michel, Fabien Gandon, Valentin Ah-Kane, Anna Bobasheva, Elena Cabrio, Olivier Corby, Raphaël Gazzotti, Alain Giboin, Santiago Marro, Tobias Mayer, Mathieu Simon, Serena Villata, and Marco

Winckler. Covid-on-the-Web : Knowledge Graph and Services to Advance COVID-19 Research. In *ISWC 2020 - 19th International Semantic Web Conference*, Athens / Virtual, Greece, November 2020.

[18] Nandana Mihindukulasooriya, Mohammad Rifat Ahmmad Rashid, Giuseppe Rizzo, Raúl García-Castro, Oscar Corcho, and Marco Torchiano. Rdf shape induction using knowledge base profiling. SAC '18, page 1952–1959, New York, NY, USA, 2018. Association for Computing Machinery.

[19] H.J. Pandit, D. O'Sullivan, and D. Lewis. Using ontology design patterns to define shacl shapes. In *WOP@ISWC*, pages 67–71, Monterey California, USA, 2018.

[20] Paolo Pareti and G. Konstantinidis. A review of shacl : From data validation to schema reasoning for rdf graphs. In *Reasoning Web*, 2021.

[21] Paolo Pareti, George Konstantinidis, Timothy J. Norman, and Murat Şensoy. Shacl constraints with inference rules. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web – ISWC 2019*, pages 539–557, Cham, 2019. Springer International Publishing.

[22] Renzo Principe, Andrea Maurino, Matteo Palmonari, Michele Ciavotta, and Blerina Spahiu. Abstat-hd : a scalable tool for profiling very large knowledge graphs. *The VLDB Journal*, 31, 09 2021.

[23] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Shacl and shex in the wild : A community survey on validating shapes generation and adoption. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 260–263, New York, NY, USA, 2022. Association for Computing Machinery.

[24] Jesse Wright, Sergio José Rodríguez Méndez, Armin Haller, Kerry Taylor, and Pouya G. Omran. Schímatos : A shacl-based web-form generator for knowledge graph editing. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 65–80, Cham, 2020. Springer International Publishing.

Session 5 : Graphes de connaissances, apprentissage, temporalité

Temporalité et graphes de connaissances : analyse théorique et enjeux pratiques

W. Charles[1], N. Aussenac-Gilles[1], N. Hernandez[1,2]

[1] IRIT- CNRS et Université de Toulouse, prenom.nom@irit.fr
[2] Université Toulouse 2 Jean Jaurès

19 mai 2023

Résumé

La représentation de faits ancrés dans une temporalité est une tâche ardue en RDF. Nous détaillons les étapes menant à la conception d'un ontologie intégrant une temporalisation des faits. Nous dégageons notamment un cadre théorique permettant de déterminer quelles propriétés temporaliser selon la conception du temps retenue, ainsi que des raisons pouvant pousser à simplifier celui-ci. Nous revenons également sur les différentes méthodes de représentation du temps et de la temporalisation des faits.

Mots-clés

Temporalité, Ontologie, Analyse Théorique, Philosophie.

Abstract

Representing temporally anchored facts in RDF is a challenging task. Here, we detail steps that can help designing an ontology integrating temporal facts. We notably propose a theoretical reference frame aiming to identify properties that are to be temporalised, as well as motives that could lead to simplifying it. We also review the various ways to represent time and fact temporalisation.

Keywords

Temporality, Ontology, Theoretical Analysis, Philosophy.

1 Introduction

Dans le contexte du Web Sémantique, RDF s'est imposé au cours des dernières années comme le standard de la représentation de connaissances [24]. Ce langage reposant sur l'usage de triplets permet de mettre en relation deux ressources qualifiées par un identifiant unique (URI) au travers d'un prédicat possédant lui-même un URI. Bien qu'extrêmement simple, ce formalisme offre de vastes possibilités de représentation. Des vocabulaires tels que OWL ou RDFS proposent une extension de la sémantique de celui-ci et accroissent son expressivité et ces possibilités de raisonnement [38]. Toutefois, si RDF permet de représenter sans difficulté des relations entre entités à un instant donné, la représentation d'entités à composante temporelle, ou plus généralement de faits temporellement dépendants y est autrement plus difficile [5]. En effet, le formalisme du triplet ne permet par exemple pas d'apposer un marqueur temporel sur une relation sans l'usage d'un mécanisme auxiliaire. De nombreuses solutions ont été mises en avant pour pallier ce manque, allant de l'usage de formalismes utilisant la sémantique RDF tels que la réification [21], jusqu'à des extensions de la syntaxe en utilisant par exemple des quadruplets au lieu de triplets [45].

Si cette limite pratique peut encore constituer un obstacle à l'intégration de données temporelles dans un graphe de connaissances, d'autres questionnements sous-tendent la représentation temporelle. En effet, dès lors que l'on commence à ajouter des composantes temporelles, il devient nécessaire de s'interroger sur la légitimité de temporaliser telle ou telle entité ou propriété. Pour ce qui est des entités, des travaux tels que ceux de l'ontologie de haut niveau DOLCE [16] introduisent deux grandes catégories d'entités temporelles : les endurants et les perdurants (similairement aux occurrents et continuants de BFO, une autre ontologie de haut niveau [4]), distinguables par la stabilité au cours de leur existence des uns, et la propension à connaître plusieurs états au cours de celle-ci des autres. Notons que cette distinction exclut *a priori* l'existence d'entités hors du temps. Or, les ontologies pratiques faisant usage de la temporalité mêlent souvent des entités temporellement marquées à d'autres qui ne le sont pas. On peut citer comme exemple l'ontologie des organisations [1], une recommandation du W3C. Bien que celle-ci inclue une dimension temporelle dans la description des organisations et des rôles au sein de celles-ci, elle n'en intègre pas pour la description des sites occupés par les entreprises. Cette pratique vise à simplifier la représentation dans son ensemble et permet de limiter le coût en stockage. Elle se justifie par l'intérêt limité que l'on peut porter à certaines ressources limitrophes au contexte visé par l'ontologie, ou par l'immuabilité locale de certains objets dans le cadre d'une représentation ciblée.

La description d'entités temporelles n'est cependant pas complète sans la temporalisation des propriétés qui les lie. Toutefois, s'il semble naturel de temporaliser certaines relation tel que l'union conjugale (`estMarieA`), d'autres telles que la relation de paternité (`pereDe`) semblent plus délicates. De par son usage courant, il serait tentant de considérer `pereDe` comme une relation atemporelle. On pourrait toutefois objecter que cette propriété n'entre en

1. https ://www.w3.org/TR/vocab-org/

existence qu'à l'instant où l'enfant cible de la relation vient au monde.
Dans cet article, nous tâcherons d'analyser les étapes et les choix de représentation à suivre lors de la conception d'une ontologie pratique intégrant une dimension temporelle. Il ne s'agit pas de donner une méthodologie définitive, mais simplement de souligner les complexités sous-jacentes à chaque étape. L'accent sera mis sur les aspects de représentation et non sur ceux de raisonnement. La section 2 propose des distinctions théoriques à réaliser lors des choix de temporalisation d'entités et de propriétés. Les considérations issues de cette section constituent la contribution principale de cet article. La section 3 traitera les aspects inhérents à la représentation du temps lui-même. La section 4 s'attachera à dresser un état de l'art des techniques employées afin de représenter la temporalité à l'aide de RDF.

2 Temporalité : que représenter ?

Avant de s'interroger sur comment représenter le temps, il est nécessaire de définir quelles ressources manipulées au sein d'une représentation doivent être ancrées dans le temps. Concernant les entités, la distinction entre endurants et perdurants, issue de considérations philosophiques classiques [32], est portée notamment par l'ontologie de haut niveau DOLCE [16]. Les entités perdurantes sont décomposées en un ensemble de parties temporelles correspondant à ce qu'est un individu à un instant donné. A un instant donné, seule une partie temporelle de l'entité perdurante est présente, ce qui signifie qu'elle n'est pas pleinement présente. Les entités endurantes, au contraire, sont pleinement présentes à tout instant de leur existence. Cela ne signifie pas qu'elles sont hors du temps, car elles peuvent avoir une période d'existence fixée. En termes concrets, il s'agit d'une distinction entre entités qui évoluent (perdurants) et qui n'évoluent pas (endurants).
Il reste toutefois à s'interroger sur la temporalisation des triplets mettant en relation ces entités. Nous inscrivons ici dans un cadre théorique les choix de représentation induisant la temporalisation de triplets. Ce cadre se fondant sur des considérations d'ordre philosophique, il est vraisemblable que la représentation pratique du temps dans le contexte d'une ontologie de domaine induise des simplifications. Dans un second temps, donc, nous mettons en exergue les raisons qui peuvent amener à simplifier ce cadre dans la pratique.

2.1 Considérations théoriques

L'interrogation qui nous occupe est la temporalité de propriétés liant deux entités. En nous appuyant sur l'exemple de `pereDe`, nous proposons une analyse des choix de représentation qui peuvent mener ou non à temporaliser la propriété. Deux dimensions sont ici considérées. La première est l'ontologie du temps retenue. On ne parle pas d'ontologie pratique mais bien d'ontologie au sens de qualification de la nature du temps. Ainsi, il existe principalement dans la philosophie trois grandes ontologies du temps [37, 8]. *L'éternalisme* affirme que le passé, le présent et le futur ont une réalité éternelle. Le *présentisme* considère

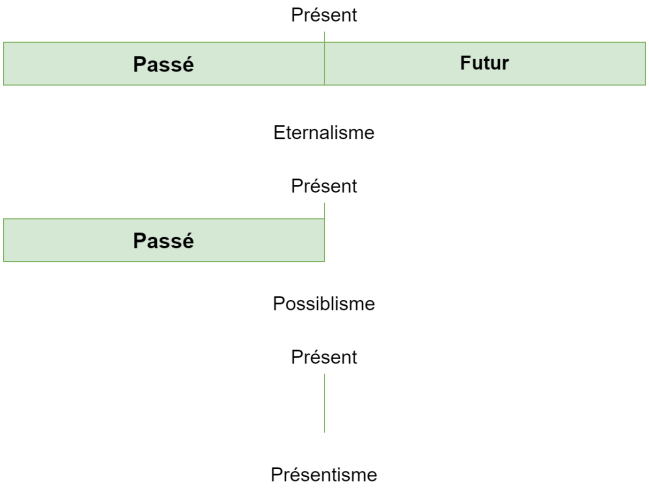


FIGURE 1 – Trois visions du temps

tout ce qui n'est pas présent est fiction, et s'appuie sur l'intuition cognitive naturelle qu'a l'individu de l'instant présent. Enfin, le *possibilisme* [37] (aussi appelé gradualisme [43] ou théorie du bloc croissant [44]) considère que le futur n'existe pas encore. Le passé y est conceptualisé comme l'ensemble des présents déjà parcourus. La figure 1 résume visuellement les composantes temporelles considérées comme existantes dans chacune des trois théories.
La seconde dimension est le monde conceptuel dans lequel s'inscrit la relation. Plus explicitement, la relation concerne-t-elle l'entité en tant que concept ou en tant qu'objet physique ? Nous allons donc analyser le statut de la propriété `pereDe` dans chacune des conceptions temporelles, selon que l'on considère qu'elle lie deux entités conceptuelles ou physiques. Dans ce paragraphe, on parlera de propriété *atemporelle* pour une propriété qui n'a pas vocation à être contextualisée dans le temps, de propriété *temporelle* pour une propriété qui n'est valable que sur des périodes temporelles de durée finie. Enfin, on parlera de propriété *semi-temporelle* pour celles dont la validité a un début, mais pas de fin.

2.1.1 Présentisme

Dans le contexte du présentisme, le passé et le futur n'ont pas de réalité. Ainsi, la notion de temporalisation n'a de sens qu'en tant que fiction mathématique [8]. Dans un contexte strictement présentiste :

- Si `pereDe` lie deux entités conceptuelles : la propriété `pereDe` est atemporelle.
- Si `pereDe` lie deux entités physiques : la relation ne peut exister que si les entités existent. Autrement dit, dans le cas de d'un humain, la relation n'existera que si les deux individus qu'elle lie sont présentement vivants. La propriété est donc atemporelle.

On notera que dans une vision présentiste, on peut néanmoins discourir sur des entités passées en considérant qu'elles correspondent à un univers fictif suivant une représentation possibiliste. Les entités représentées ne sont alors plus les entités réelles mais des projections dans un univers

Modèle temporel	Entités Conceptuelles	Entités Physiques
Présentisme	Atemporelle	Atemporelle si existante
Possibilisme	Semi-temporelle	Temporelle ou Semi-Temporelle
Eternalisme	Atemporelle	Temporelle

TABLE 1 – Nature des propriétés selon l'ontologie du temps et l'aspect des entités considérés

fictif.

2.1.2 Possibilisme

Le possibilisme établit l'existence du passé et du présent. Dans ce contexte :

- Si `pereDe` lie deux entités conceptuelles : la propriété `pereDe` est semi-temporelle. En effet, il existe un présent précédant la naissance de l'individu où même le concept de l'individu n'existe pas. La propriété `pereDe` n'est donc valable qu'à compter du moment où la conceptualisation de la réalisation de la propriété est possible.
- Si `pereDe` lie deux entités physiques : la propriété est temporelle, ou semi-temporelle selon que les individus concernés soient tous deux en vie ou non.

2.1.3 Eternalisme

L'éternalisme considère l'existence perpétuelle du passé, présent et futur. Nous en tirons les conséquences suivantes :

- Si `pereDe` lie deux entités conceptuelles : la propriété est atemporelle. En effet, le futur existant, la relation conceptuelle existe également à tout instant.
- Si `pereDe` lie deux entités physiques : la propriété est temporelle, car elle lie deux entités physiques qui n'existent qu'à une période donnée. On ne peut pas toujours savoir à quel instant elle prendra fin, mais elle demeure temporelle car le futur est déjà partie intégrante de la réalité.

La table 1 récapitule la section 2.1.

2.2 Considérations pratiques

En pratique toutefois, ces considérations sont parfois écartées de sorte à obtenir une représentation davantage adaptée aux besoins du domaine représenté. Nous avons identifié deux cas distincts permettant de justifier l'omission de la temporalisation de certaines entités, tous liés à la notion de domaine de représentation.

2.2.1 Notion limitrophe au cadre de l'ontologie

Le premier cas englobe les situations où la temporalité est supprimée de sorte à ne pas alourdir la représentation plus que nécessaire. Formellement, pour un triplet (s, p, o), légitimement temporel dans le cadre de représentation retenu par l'ontologie, on distingue plusieurs options ;

- Le triplet n'est pas temporalisé, et o non plus. C'est le cas de l'ontologie des organisations, mentionnée dans l'introduction. Ce cas appauvrit la représentation temporelle de s, mais cela peut être motivé par le peu d'importance que l'on accorde à certaines propriétés.
- Le triplet est temporalisé, mais o ne possède aucune représentation temporelle. Dans le contexte de la représentation de territoires évolutifs par exemple, il est possible de représenter un lien temporellement marqué avec l'individu régissant un territoire, sans pour autant donner une représentation temporelle à l'individu en question. La représentation de celui-ci n'est pas ce qui nous occupe dans ce contexte. Bien que ne limitant pas la représentation temporelle de s, cette solution complique la réutilisabilité dans le cas de la confrontation à un jeu de données inscrivant o dans une temporalité (typiquement si le jeu de données fait usage d'un modèle *fluents* décrit dans la section 4.2.1).
- Le triplet n'est pas temporalisé, mais o l'est. Ce cas de figure peut se présenter quand la propriété p n'a guère d'importance dans notre contexte. Dans des travaux précédents [12], nous avions également mis en avant l'usage de cette solution pour alléger la représentation en temporalisant implicitement la propriété p. Cette dernière était alors considérée valide à l'intersection des intervalles marquant l'existence des objets temporels o et s.

2.2.2 Immuabilité locale au cadre d'étude

Certaines entités peuvent être considérées comme localement immuables du fait du contexte retenu pour l'ontologie. Par exemple, dans le contexte d'une ontologie de représentation de territoires contemporains [15], il est possible de considérer les éléments topographiques tels que les montagnes comme des éléments immuables. En effet, bien qu'ils évoluent au fil du temps et soient voués à disparaître, la période couverte par le discours ne justifie par leur temporalisation au sein de l'ontologie.

3 Représentation du temps

Les considérations décrites dans la section précédente permettent de trancher quant aux composantes de l'ontologie susceptibles d'être plongées dans le temps. Avant de représenter la temporalité de celles-ci, il convient toutefois de s'interroger sur le formalisme temporel à retenir pour représenter le temps en lui-même.

3.1 Représentation théorique du temps

D'un point de vue ontologique, plusieurs approches ont été mises en avant pour représenter et raisonner sur le temps. Cette représentation est polarisée entre les représentations à base de points et celles à base d'intervalles, les premières étant fondées sur l'intuition mathématique et physique du temps, tandis que l'autre repose davantage sur l'expression du temps en langage naturel [3].

3.1.1 Le temps en points

La première option consiste à considérer le temps comme un ensemble ordonné de points [9, 31]. Cette approche permet de représenter des intervalles temporels, soit à l'aide d'ensembles de points [31] (on discrétise alors le temps),

soit à l'aide de paires (a, b) de points ordonnés formant un intervalle $[a, b]$. Dans ce dernier cas, l'intervalle est matérialisé par un instant de début a et un instant de fin b [9]. Bien que ces approches aient leurs intérêts qui justifient leur usage aujourd'hui encore (cf Table 2), elles peinent à représenter les *Points de rupture* [29], instants auxquels une propriété change de valeur logique. En effet, supposons une propriété t vraie sur un intervalle $[p1, p]$ et fausse sur un intervalle $[p, p2]$. Que se passe-t-il alors au point p ? La propriété est-elle vraie ? Fausse ? Vraie et fausse ? Ni vraie ni fausse ?

3.1.2 Intervalles temporels

Cette limite des modèles à base de points pousse à la mise en place d'un modèle basé sur des intervalles. L'algèbre temporelle d'Allen [1] définit un ensemble de 13 relations permettant de décrire et raisonner sur des intervalles. Cette représentation est fondée davantage sur la représentation des liens entre intervalles que sur un ancrage temporel de ceux-ci. Autrement dit, le raisonnement à l'aide d'intervalles d'Allen s'effectue à l'aide des relations définies par Allen et des lois de composition proposées, et ne nécessite pas d'imposer des dates à ceux-ci. Ainsi, pour le problème précédent, on a alors t vraie sur un intervalle $i1$ et fausse sur un intervalle $i2$ avec $meets(i1, i2)$ (signifiant que les intervalles $i1$ et $i2$ sont directement consécutifs). Le problème du point de jonction est évacué. Toutefois quand on parle d'intervalle ici, il faut bien comprendre qu'on ne parle pas de l'objet mathématique dont on a l'habitude, mais d'un objet défini par la sémantique de l'algèbre d'Allen. Ainsi, la version initiale ne rend pas possible la représentation d'intervalles de durée nulle (ex : $[a, a]$). Cet aspect est argumenté par le fait qu'en pratique, on peut ramener un instant à un intervalle de temps de durée très faible. Cet argument est battu en brèche dans le contexte de l'application des lois de la physique [29]. Le contre-exemple classique est la description de l'instant où la vitesse d'une balle lancée en l'air est nulle, que la physique identifie comme étant un instant et non un intervalle de durée très faible. Pour pallier ce manque, Allen proposera plus tard [2] la notion de moment qui constitue un intervalle insécable, ainsi que la notion de nid, qui permet de définir des concepts équivalents aux points en mettant en avant une relation d'ordre complète sur des ensembles d'intervalles. Toutefois, cette représentation est ardue à utiliser en pratique.

3.1.3 Une combinaison points et intervalles

[29] tente d'unifier le meilleur des deux mondes en proposant une théorie du temps mélangeant points et intervalles. Il met en place un système mathématique à base de relations étendant celles de l'algèbre d'Allen. A l'instar de celle-ci, le modèle proposé consiste davantage en une représentation fondée sur le langage naturel et les relations liant les objets temporels qu'en une représentation du temps traditionnelle faisant usage de marqueurs temporels indexables par des entiers.

3.2 Représentation pratique du temps

Dans le cadre des recherches sur l'intelligence artificielle, de nombreux modèles de représentation temporelle ont été proposés. Les formalismes présentés dans la section précédente ne s'attachent toutefois qu'à décrire les relations entre objets temporels.

3.2.1 Intervalles mathématiques

En pratique, de nombreux modèles font usage d'intervalles mathématiques classiques. Dans le contexte des modèles à base de points, un intervalle I est défini par deux points a et b ($a < b$). Certains modèles [6, 25, 19, 14] utilisent simplement ce formalisme. Or, en mathématiques, on distingue pour les intervalles des bornes fermées et ouvertes, indiquant si la borne concernée appartient ou non à l'intervalle. Notons que cette approche présente un défaut dans un contexte de représentation informatique. En effet, la représentation machine implique nécessairement une discrétisation du temps, tandis que la qualification des bornes évoquée précédemment se fonde sur la représentation d'intervalles continus. Néanmoins, cette représentation est utilisée en pratique, (parfois en conjonction avec l'algèbre d'Allen [25]), car elle permet notamment de lever le problème du point de rupture. Dans la pratique on distingue deux cas d'usage :

— La spécification des bornes est libre [6, 25, 19]. Il est donc de la responsabilité de l'utilisateur de construire un modèle cohérent.
— Une borne est fixée comme étant ouverte (par exemple la borne supérieure dans [14]). Notons que dans ce cas, le cadre de représentation tranche le problème du point de rupture, en indiquant qu'en ce point la propriété prend la valeur qu'elle aura par la suite.

3.2.2 Ontologies pratiques de représentation du temps

En pratique, la représentation du temps dans les ontologies est souvent réalisée à l'aide de OWL-Time [2]. Sur 19 ontologies du temps (portant le tag "Time") présentes sur Linked Open Vocabularies (LOV [3]) au moment de la rédaction de cet article, OWL-Time est de loin la plus utilisée avec 50 liens entrants (cf figure 2). Cette ontologie constitue un standard de représentation du Web Sémantique. Elle couvre une vaste palette de possibilités de représentations explicites du temps, aussi bien du point de vue d'une représentation proche d'usages usuels (notion de jour, mois, années, calendrier ...) que mathématique (usage de points temporels pour ancrer les éléments dans le temps). Elle propose également une implémentation de l'algèbre d'Allen pour les intervalles. OWL-Time fait notamment usage des littéraux `xsd:date` et `xsd:dateTime` qui sont parfois utilisés directement au sein d'ontologies afin de représenter certains aspects temporels sans le truchement d'OWL-Time [41]. Toutefois le défaut principal d'OWL-Time est que les primitives proposées ne permettent pas de représenter explicitement le flou, i.e. l'imprécision et/ou l'incertitude liées à

2. https ://www.w3.org/TR/owl-time/
3. https ://lov.linkeddata.es/dataset/lov

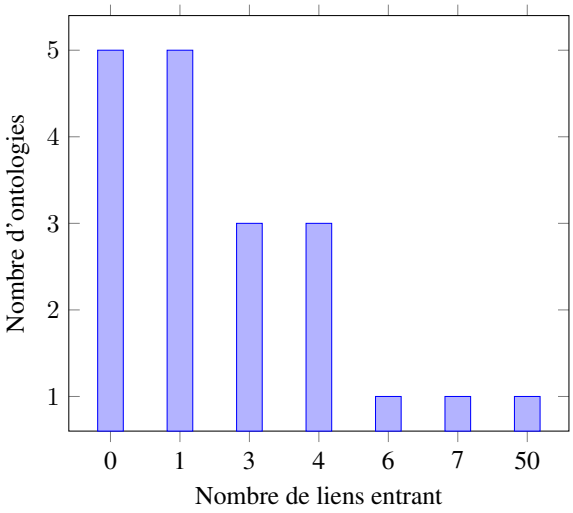


FIGURE 2 – Répartition du nombre de liens entrants pour les ontologies portant le tag "Time" référencées sur le LOV

cette dimension.

3.2.3 Incertitude et imprécisions temporelles

Dans OWL-Time la seule représentation d'une quelconque incertitude ou imprécision est implicite. En effet, lors de la spécification d'un `time:Instant`, la valeur précisée peut être simplement un mois ou une année, permettant de maintenir une imprécision quant à la date effective de l'instant en question. Nombre de modèles plus expressifs existent. Tout d'abord, il existe des modèles reposant sur la pondération de valeurs temporelles, associés à un modèle de raisonnement sur des données incertaines (exemple : logique de Markov [25]). Intuitivement, cette pondération peut être assimilée à la probabilité qu'un fait se soit produit à une date donnée. Toutefois, que ce soit pour [25], [45] ou encore [14], la représentation d'une temporalité vague ne porte pas sur la composante temporelle, mais sur le triplet temporalisé (cf section 4.1). Il est par exemple impossible de déclarer qu'on est certain qu'un fait ait eu lieu sans avoir une idée de sa date exacte.
Comme mentionné précédemment, la représentation de l'algèbre d'Allen repose davantage sur les relations entre intervalles que sur leur datation. Ainsi, ces relations peuvent être utilisées pour représenter certains types d'informations temporelles vagues. Par exemple, elles permettent de décrire des intervalles vagues comme *"Un moment X dans l'après-midi."* : *during(ApresMidi, X)* sans avoir à définir précisément ce qu'est l'après-midi (c'est simplement un intervalle après le matin, et avant le soir). Elles ne permettent cependant pas de représenter des périodes telles que *"Un moment X aux alentours de 13h"* sans définir un intervalle des "alentours".
Une approche commune dans le contexte des humanités numériques, car recommandée par le modèle conceptuel CIDOC-CRM [10] est l'intervalle à quatre points. La borne de début d'intervalle d y est remplacée par une paire (d_{min}, d_{max}) formant un intervalle dans lequel le début effectif de la période décrite se trouvera. Il en va de même pour la borne de fin. Concernant *"Un moment X aux alentours de 13h"*, l'intervalle à quatre points permet d'être moins précis quant à la définition de l'intervalle des "alentours", mais requiert toujours une définition de celui-ci.
Dans ce même contexte, on trouve également des ontologies ou des *gazeteers* qui utilisent pour marqueur temporel des périodes déterminées par des civilisations [18] ou géologiques [13]. Une autre forme de temporalité vague consiste à représenter des nombres d'occurrences au cours d'une période donnée. Cette méthode est appelée *triplet indéterminé* dans [39]. La partie temporelle du modèle SEAS [28] met également en avant ce type de représentations, accompagnée de méthodes permettant d'effectuer des calculs sur les occurrences.
Enfin, Temporal OWL propose la notion de temporalité anonyme qui permet d'indiquer qu'un fait se produit à un instant donné sans pour autant devoir spécifier une quelconque information sur la période en question [22].

3.2.4 Autres marqueurs spéciaux

La représentation de données temporelles se heurte au fait que les connaissances que l'on peut avoir s'arrêtent souvent à l'instant présent. Ainsi, plusieurs formalismes proposent la mise en place d'un marqueur matérialisant l'instant présent [6, 21, 39, 33, 19]. On trouve globalement trois marqueurs pour réaliser ceci : `NOW`, `UC` (pour *Until Changed*) et ∞. Si certains modèles comme Temporal RDF font emploi de plusieurs marqueurs auxquels ils attribuent des sémantiques distinctes, d'autres [19] en utilisent un seul pour couvrir plusieurs sens. Il est donc difficile de donner une sémantique universelle à ces trois marqueurs. Comme souligné par [19], toutefois, ces marqueurs sont hérités des bases de données relationnelles temporelles, où `NOW` et ∞ sont utilisés afin de représenter des connaissances valables jusqu'à l'instant présent ou à l'infini, tandis qu'`UC` est utilisé dans un contexte transactionnel. Le symbole ∞, issu des mathématiques, est plus délicat car il est utilisé pour représenter alternativement l'instant présent ou le fait que l'intervalle de validité d'une propriété est étendu jusqu'à l'infini.

4 Représenter la temporalité

Enfin, afin de temporaliser les faits, il est nécessaire de choisir un formalisme de représentation pour les graphes de connaissances adapté aux besoins de temporalisation identifiés selon les principes énoncés section 2. Nous dressons ici un état de l'art des méthodes utilisés afin de lier les ressources aux marqueurs temporels issus d'un formalisme de la section 3. Un état de l'art plus détaillé est proposé dans [46]

4.1 Marquage temporel des propriétés

Une approche courante pour représenter la temporalité au sein d'un graphe RDF consiste à apposer un marqueur temporel à un triplet, indiquant sa période de validité [6, 21, 49]. Toutefois, le formalisme RDF ne permet pas à

lui seul cette représentation. Des mécanismes conçus pour l'ajout de métadonnées peuvent alors être utilisés [45]. La figure 3 présente la représentation temporelle de deux triplets ayant la même période de validité à l'aide de divers mécanismes que nous allons détailler ici.

4.1.1 Réification standard et Temporal RDF

La réification standard [4] est un mécanisme introduit par le W3C permettant de représenter des méta-propriétés [30]. Un triplet y est représenté sous la forme d'une instance de la classe `rdf:statement` auquel on adjoint les composantes du triplet à l'aide des propriétés `rdf:subject`, `rdf:predicate` et `rdf:object`. La figure 3a présente un exemple de réification. L'une des premières approches dédiées à la représentation temporelle dans RDF, Temporal RDF [21, 22], fait appel au mécanisme de réification, en utilisant un vocabulaire qui lui est propre. Cette approche traite la temporalité en adjoignant à tout triplet *(s, p, o)* une composante temporelle *t* indiquant la période de validité du triplet. En pratique nombre de formalismes de temporalisation des triplets font usage de la réification RDF [6, 25] bien que certains définissent en sus une syntaxe propre à leur modèle de représentation temporelle, complétée d'un formalisme permettant la conversion vers le modèle RDF en faisant usage de la réification.

4.1.2 Réification standard : alternatives

Le défaut majeur de la réification RDF standard est sa lourdeur syntaxique, qui amène souvent les auteurs à définir leur propre syntaxe afin d'alléger la représentation. Ainsi, de nombreuses alternatives à la réification standard ont été mises en avant, chacune présentant des avantages et des inconvénients dans le cadre de la représentation temporelle.

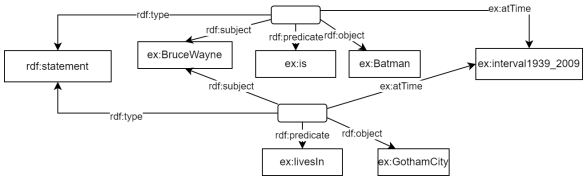
- *Les relations n-aires* [5] permettent de représenter des relations de cardinalité supérieure à la binarité imposée par le triplet. Elles consistent à matérialiser la relation par une entité à laquelle sont rattachées diverses propriétés. Bien que plus légère syntaxiquement que la réification, l'utilisation de relations n-aires nécessite la définition préalable de classes dont les entités matérialisant les relations seront des instances, repoussant une partie de la complexité dans la conception de l'ontologie associée. Dans le contexte de l'adjonction d'un marqueur temporel, deux options d'utilisation des relations n-aires sont possibles. La première, illustrée figure 3b, consiste à créer une classe de relation pour chaque propriété à laquelle on souhaite adjoindre un marqueur temporel. La seconde, illustrée figure 3c, ne crée qu'une seule classe de relation qui correspond aux propriétés temporelles, sur laquelle viendront se greffer l'ensemble des propriétés. Ainsi, plusieurs relations vérifiées à une même période seront regroupées sur une unique relation n-aire. Cette seconde utilisation, bien que plus compacte pour le stockage, semble peu intuitive, et présente également le défaut de détourner les relations n-aires de leur sémantique première. En effet, là où les relations n-aires escomptent une valeur pour chacune des propriétés définies pour la classe des entités représentant la relation, le nombre de valeurs effectivement renseignées est ici variable.
- *Les graphes nommés* [11] sont un formalisme visant à permettre de faire référence à un ensemble de triplets, regroupés au sein d'un graphe, comme ressource RDF (cf figure 3d). Dans le contexte de la temporalisation, ce mécanisme n'a d'intérêt que si un grand nombre de triplets partagent la même période de validité. La représentation de tels graphes implique que dans le cas où un triplet serait valide à des périodes non connexes, celui-ci devrait être dupliqué dans chacun des graphes temporellement marqués pour lequel il est vérifié.
- *RDF-star* [23] est une extension de RDF visant à permettre de considérer un triplet comme une ressource (figure 3e). Il s'agit d'une forme de réification alternative à celle définie dans la sémantique de RDF, qui s'accompagne de syntaxes simplifiées pour alléger l'écritures de requêtes et de graphes de connaissances. Bien que naturel pour l'ajout d'une composante temporelle à un triplet, ce formalisme est encore peu utilisé. Sa standardisation par le W3C est toutefois en cours.
- *Les Singleton Properties* [34] (figure 3f) sont un mécanisme consistant à utiliser une réalisation unique de la propriété dans chaque triplet. Ainsi, si on utilise une propriété *p* dans deux triples différents, ces derniers utiliseront en pratique des propriétés *p#1* et *p#2*. Ces propriétés singletons étant utilisées dans un unique triplet, elles peuvent ainsi servir à le désigner. Une composante temporelle peut donc par exemple y être apposée. A noter qu'une extension des premier travaux [35] permet de préserver l'inférence lors de l'usage de propriétés singletons.
- Enfin, les méthodes d'annotation telles que aRDF [45] permettent l'ajout de la composante temporelle sous forme d'annotation. Il existe notamment des formalismes d'annotation, tels que RDFt [48], qui se concentrent exclusivement sur l'annotation temporelle. Toutefois, ces mécanismes reposent sur des extensions de la syntaxe RDF, ce qui complique leur réutilisabilité.

Il est à noter que nombre de formalismes de représentation du temps [6, 21, 33, 25] fondent leur sémantique sur un formalisme abstrait reposant sur l'adjonction d'une composante au triplet, certains allant même jusqu'à s'abstraire de toute implantation [33]. Ainsi, si les réalisations proposées s'appuient sur la réification, il semble pertinent de considérer des alternatives à celle-ci dans le but de simplifier le stockage [5].

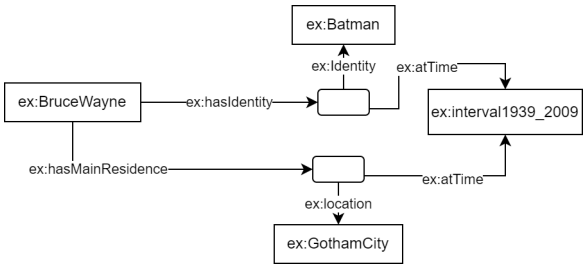
4.2 Représentation de séries temporelles

L'estampillage temporel de triplets constitue une solution pour représenter des connaissances ancrées dans une temporalité fixée. Une autre solution consiste à porter la com-

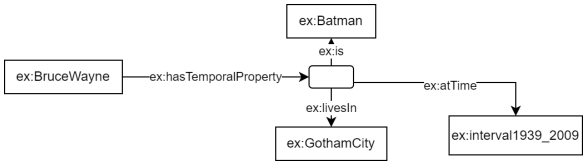
4. `https://www.w3.org/wiki/RdfReification`
5. `https://www.w3.org/TR/swbp-n-aryRelations/`



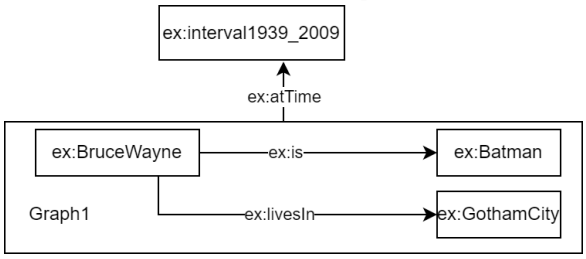
(a) Réification standard



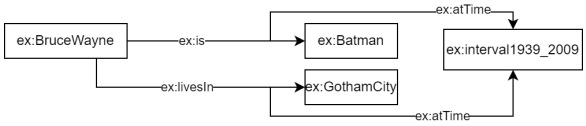
(b) Relation n-aire : option 1



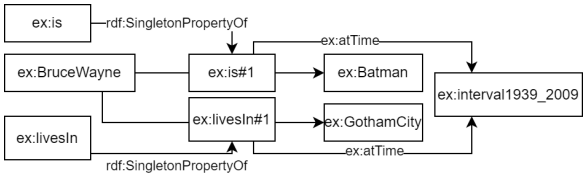
(c) Relation n-aire : option 2



(d) Graphe nommé



(e) RDF-star



(f) Singleton Property

FIGURE 3 – Réification standard et alternatives pour représenter la temporalité

posante temporelle non sur les triplets mais sur les entités représentées. On parle de sérialisation, car les entités sont alors représentées par une série d'états temporels. [20] propose deux modèles de sérialisation : les ontologies SPAN et SNAP. Les premières décrivent le cas où chaque entité déroule ses parties temporelles au cours du temps dans un unique graphe de connaissances. La seconde correspond au cas où l'on décrit une série de "snapshots" représentant l'état complet des connaissances à un instant donné.

4.2.1 Le modèle fluents

Le modèle fluents [47] est mis en avant pour représenter des entités perdurantes. Il repose sur un mécanisme relativement simple consistant à associer à chaque entité plusieurs parties temporelles, représentant son état à diverses périodes. Cette approche a par ailleurs été étendue afin de représenter une entité dans un contexte de nature quelconque (pas nécessairement temporel) [17]. Toutefois, de nombreuses questions sont soulevées en pratique. Tout d'abord, on notera que, selon DOLCE, des entités perdurantes sont amenées à coexister avec des entités endurantes, ce qui peut complexifier le graphe. Une autre complexité porte sur l'usage de propriétés que l'on qualifiera d'atemporelles. La date de naissance d'un individu, par exemple, ne dépend pas de son état, et pourra ainsi être attachée non pas à chacune des parties temporelles, mais bien à l'entité porteuse de l'identité de l'individu. Enfin, outre l'expression de requêtes qui est considérablement alourdie, le principal problème généré par cette approche est la prolifération de triplets, comme avancé par [5]. Tout d'abord, la notion de partie temporelle implique que l'ensemble des propriétés permettant de qualifier un état doivent être décrites pour chacun des états. Si la description porte sur un être humain, et que les parties temporelles donnent sa taille et son nom, pour chaque évolution de la taille, il sera nécessaire d'ajouter un nouveau triplet rattachant la nouvelle partie temporelle au nom de l'individu même si celui-ci reste identique. Un autre facteur de prolifération est la mise en relation entre diverses entités perdurantes. En effet, si une partie temporelle de A est en lien avec B, mais que sur la période couverte par ladite partie de A, B possède plusieurs parties temporelles, il peut être nécessaire de fragmenter la partie temporelle de A pour être en adéquation avec B. Cette complexité peut être réduite en considérant que la validité d'une relation entre deux parties temporelles se trouve à l'intersection des périodes de validités de celles-ci, comme mis en avant par des travaux précédents [12].

4.2.2 Séries de graphes

L'usage de séries de graphes pour représenter la temporalité consiste à représenter l'ensemble des connaissances valables à un instant donné sous forme d'un graphe de connaissances atemporel. L'ensemble des graphes ainsi produits permet de représenter l'évolution des faits au cours du temps [26]. Bien que reposant sur la représentation d'un état de fait à un instant, cette méthode permet de représenter des entités perdurantes en considérant les entités représentées dans chacun des graphes successifs comme autant de parties temporelles. Cette méthode est également utilisée

pour représenter l'évolution de l'état des connaissances sur un domaine, avec une notion de version de graphe [36].
La problématique principale des approches par séries de graphes est qu'elles requièrent un certain synchronisme dans l'évolution des faits. En effet, dans le cas où on décrit un grand nombre d'entités susceptibles d'évoluer, ce mécanisme a pour défaut de nécessiter la duplication de l'entièreté du graphe pour chaque modification individuelle, ce qui rend son usage lourd dans le cas où les entités évoluent fréquemment et indépendamment les unes des autres. [36] propose une approche pour corriger ce défaut. En mettant en avant que le triplet est un élément insécable de tout graphe RDF et que celui-ci ne peut être qu'ajouté ou supprimé (l'édition correspondant à une suppression suivie d'un ajout), leur approche se base sur la représentation des modifications subies d'un graphe à l'autre.
Dans le contexte de séries de graphes, une difficulté est l'identification des ressources décrites d'un graphe à l'autre. L'idée sous-jacente à ces séries de graphes est généralement qu'une entité connaissant une évolution change d'identité. Autrement dit, il n'y a pas de raison qu'une ressource décrite dans deux graphes successifs possède le même URI dans chacun des graphes. En pratique d'ailleurs, l'usage d'un unique URI peut être problématique car la combinaison de deux graphes peut conduire à des connaissances contradictoires (car les connaissances au sein des graphes ne sont pas temporalisées). Afin de pouvoir représenter l'équivalence ou la succession des ressources décrites, des approches telles que le Change Bridge du projet SAMPO [26] permettent de lier une à une les ressources des différents graphes, faisant apparaître explicitement la ligne temporelle d'une ressource. On trouve également des approches hybrides, combinant le modèle Fluents et les Change Bridge [7].

4.3 Temporalité et formalisme de représentation

Du fait des enjeux que présente la représentation temporelle, de nombreux travaux se sont attachés à définir un cadre de représentation. La table 2 dresse une liste (non-exhaustive) de ces formalismes en comparant leurs possibilités de représentation (et non de raisonnement). A l'exception de TA-RDF [40] qui s'approche du modèle fluents, l'ensemble des formalismes présentés utilisent une mécanique d'annotation de triplet. Nombre d'entre eux complètent leur formalisme d'une syntaxe propre accompagnée d'un algorithme permettant de traduire les connaissances ainsi représentées en RDF classique [6, 39, 45, 25]. Ces modèles sont par ailleurs souvent complétés par un langage de requêtage propre généralement dérivé de SPARQL [6, 40]. Concernant le formalisme retenu, on constate une dominance des approches par points et par intervalles mathématiques, qui ont pour avantage de faciliter l'ancrage dans une temporalité précise. Enfin, les enjeux de représentation tels que l'instant présent ou la représentation de périodes temporelles vagues sont intégrés de manière disparate au sein des diverses approches.

Approche	Marqueur Présent	Temporalité vague	Représentation du temps
stRDF[6]	✓		Intervalles mathématiques
Temporal RDF[21, 22]	✓	Temporalité Anonyme	Point
tRDF[39]	✓	Triplet indéterminé	Point
Annotated RDF[49]			Intervalles fermés + Opérateurs
aRDF[45]		Pondération possible	Variable, mais nécessite une relation d'ordre
[33]	✓		Point
[25]		Pondération (Markov)	Allen + Intervalles mathématiques
[19]	✓		Intervalles mathématiques
TA-RDF [40]			Point
[14]		Pondération	Intervalles ouverts

TABLE 2 – Comparaison de différents formalismes de représentation temporelle

5 Conclusion

Lors de la conception d'une ontologie intégrant une temporalisation des faits, il est nécessaire de s'interroger sur la légitimité de temporaliser tel ou tel fait selon la conception du temps retenue. Si certains travaux [42] suggèrent que l'éternalisme serait l'alternative la plus adaptée dans un contexte physique, d'autres [44] mettent en avant l'utilité du possibilisme dans le contexte des humanités numériques par exemple. Selon le cadre de représentation de l'ontologie développée, il est donc bon de s'interroger sur la représentation la plus adaptée. Ceci étant acté, une représentation des composantes temporelles doit être choisie en fonction du contexte. Dans celui des humanités numériques par exemple, il sera certainement nécessaire de retenir un formalisme considérant des composantes temporelles imprécises/incertaines [10]. Dans certaines circonstances, la temporalisation pourra s'effectuer en se concentrant sur la représentation exclusive d'évènements [27]. Enfin, il sera important de s'interroger sur la technique à utiliser afin de lier les faits et les composantes temporelles exprimées à l'aide du formalisme retenu. Dans cet article, nous fournissons des éléments qualitatifs permettant d'orienter le choix selon la nature des données représentées. Des travaux actuellement en cours visent à comparer quantitativement ces divers formalismes, notamment concernant l'impact du nombre de propriétés moyen liant les entités, la propension d'une entité à évoluer, le degré de corrélation entre le changement de plusieurs propriété d'une même entité ainsi que la syn-

chronicité avec laquelle les différentes entités évoluent.

Références

[1] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11) :832–843, 1983.

[2] James F Allen and Patrick J Hayes. A common-sense theory of time. In *IJCAI*, volume 85, pages 528–531. Citeseer, 1985.

[3] James F Allen and Patrick J Hayes. Moments and points in an interval-based temporal logic. *Computational Intelligence*, 5(3) :225–238, 1989.

[4] Robert Arp, Barry Smith, and Andrew D Spear. *Building ontologies with basic formal ontology*. Mit Press, 2015.

[5] Sotiris Batsakis and Euripides GM Petrakis. Sowl : a framework for handling spatio-temporal information in owl 2.0. In *Int. Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 242 –249. Springer, 2011.

[6] Konstantina Bereta, Panayiotis Smeros, and Manolis Koubarakis. Representation and querying of valid time of triples in linked geospatial data. In *The Semantic Web : Semantics and Big Data : 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings 10*, pages 259 –274. Springer, 2013.

[7] Camille Bernard, Marlène Villanova-Oliver, Jerome Gensel, and Hy Dao. Modeling changes in territorial partitions over time : Ontologies tsn and tsn-change. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing Pages (SAC '18)*, page 866–875, 04 2018.

[8] Fabien Besnard. Time of philosophers, time of physicists, time of mathematicians. *arXiv preprint arXiv :1104.4551*, 2011.

[9] Bertram C Bruce. A model for temporal references and its application in a question answering program. *Artificial intelligence*, 3 :1–25, 1972.

[10] George Bruseker, Nicola Carboni, and Anaïs Guillem. Cultural heritage data management : the role of formal ontology and cidoc crm. *Heritage and Archaeology in the Digital Age*, pages 93–131, 2017.

[11] Jeremy J Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs. *Journal of Web Semantics*, 3(4) :247–267, 2005.

[12] William Charles, Nathalie Hernandez, and Nathalie Aussenac-Gilles. Hht : An approach for representing temporally-evolving historical territories. In *Proc. of Extended Semantic Web Conference 2023, Hersonissos, Gr.*, page to Appear, 2023.

[13] Simon JD Cox and SM Richard. A geologic timescale ontology and service. *Earth Science Informatics*, 8 :5–19, 2015.

[14] Maximilian Dylla, Mauro Sozio, and Martin Theobald. Resolving temporal conflicts in inconsistent rdf knowledge bases. *Datenbanksysteme für Business, Technologie und Web (BTW)*, 2011.

[15] Amer Ezoji and Nada Matta. Territorial knowledge ontology as a guide for the identification of resource of the territory toward sustainability. In *Proceedings of the Design Society : International Conference on Engineering Design*, volume 1, pages 3391–3400. Cambridge University Press, 2019.

[16] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. In *Knowledge Engineering and Knowledge Management : Ontologies and the Semantic Web : Proc. of 13th International Conference, EKAW 2002 Sigüenza, Spain, Oct. 1–4, 2002*, pages 166–181. Springer, 2002.

[17] José Miguel Giménez García. *Formalizing, Capturing, and Managing the Context of Statements in the Semantic Web*. Theses, Université de Lyon, July 2022.

[18] Patrick Golden and Ryan Shaw. Nanopublication beyond the sciences : the periodo period gazetteer. *PeerJ Computer Science*, 2 :e44, 2016.

[19] Fabio Grandi. Multi-temporal rdf ontology versioning. In *IWOD@ ISWC*, 2009.

[20] Pierre Grenon and Barry Smith. Snap and span : Towards dynamic spatial ontology. *Spatial Cognition & Computation*, 4 :104 – 69, 2004.

[21] Claudio Gutierrez, Carlos Hurtado, and Alejandro Vaisman. Temporal rdf. In *The Semantic Web : Research and Applications : Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29–June 1, 2005. Proceedings 2*, pages 93–107. Springer, 2005.

[22] Claudio Gutierrez, Carlos A Hurtado, and Alejandro Vaisman. Introducing time into rdf. *IEEE Transactions on Knowledge and Data Engineering*, 19(2) :207–218, 2006.

[23] Olaf Hartig. Foundations of rdf* and sparql* : (an alternative approach to statement-level metadata in rdf). In Juan Reutter and Divesh Srivastava, editors, *AMW 2017 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay*, volume 1912, 2017.

[24] Pascal Hitzler. A review of the semantic web field. *Communications of the ACM*, 64(2) :76–83, 2021.

[25] Jakob Huber. Temporal reasoning for rdf (s) : A markov logic based approach. Technical report, University of Mannheim, G., 2014.

[26] Tomi Kauppinen and Eero Hyvönen. Modeling and reasoning about changes in ontology time series. *Ontologies : A handbook of principles, concepts and applications in information systems*, pages 319–338, 2007.

[27] Gerard Kuys and Ansgar Scherp. Representing persons and objects in complex historical events using the event model f. *Journal of Open Humanities Data*, 8, 2022.

[28] Maxime Lefrançois, Jarmo Kalaoja, Takoua Ghariani, and Antoine Zimmermann. The SEAS Knowledge Model. Research Report Deliverable 2.2, ITEA2 12004 Smart Energy Aware Systems, January 2017.

[29] Jixin Ma. Ontological considerations of time, meta-predicates and temporal propositions. *Applied Ontology*, 2(1) :37–66, 2007.

[30] Frank Manola, Eric Miller, Brian McBride, et al. Rdf primer. *W3C recommendation*, 10(1-107) :6, 2004.

[31] Drew McDermott. A temporal logic for reasoning about processes and plans. *Cognitive science*, 6(2) :101–155, 1982.

[32] Neil McKinnon. The endurance/perdurance distinction. *Australasian Journal of Philosophy*, 80(3) :288–306, 2002.

[33] Boris Motik. Representing and querying validity time in rdf and owl : A logic-based approach. *Journal of Web Semantics*, 12 :3–21, 2012.

[34] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like rdf reification ? making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770, 2014.

[35] Vinh Nguyen, Olivier Bodenreider, Krishnaprasad Thirunarayan, Gang Fu, Evan Bolton, Núria Queralt Rosinach, Laura I Furlong, Michel Dumontier, and Amit Sheth. On reasoning with rdf statements about statements using singleton property triples. *arXiv preprint arXiv :1509.04513*, 2015.

[36] Damyan Ognyanov and Atanas Kiryakov. Tracking changes in rdf(s) repositories. In *Knowledge Engineering and Knowledge Management : Ontologies and the Semantic Web : 13th International Conference, EKAW 2002 Sigüenza, Spain, October 1–4, 2002 Proceedings*, pages 373–378. Springer, 2002.

[37] Daniel Peterson and Michael Silberstein. *Relativity of simultaneity and eternalism : In defense of the block universe*. Springer, 2010.

[38] Axel Polleres, Aidan Hogan, Renaud Delbru, and Jürgen Umbrich. Rdfs and owl reasoning for linked data. In *Reasoning Web. Semantic Technologies for Intelligent Data Access : 9th International Summer School 2013, Mannheim, G., July 30–Aug. 2, 2013. Proceedings*, pages 91–149. Springer, 2013.

[39] Andrea Pugliese, Octavian Udrea, and VS Subrahmanian. Scaling rdf with time. In *Proceedings of the 17th international conference on World Wide Web*, pages 605–614, 2008.

[40] Alejandro Rodrıguez, Robert McGrath, Yong Liu, James Myers, and I Urbana-Champaign. Semantic management of streaming data. In *Proc. of the 2nd International Workshop on Semantic Sensor Networks (SSN09), collocated with the 8th Int. Semantic Web Conference (ISWC-2009), Washington DC, USA*, volume 522 of *CEUR Workshop Proceedings*, pages 80–95, 2009.

[41] Anisa Rula, Matteo Palmonari, Andreas Harth, Steffen Stadtmüller, and Andrea Maurino. On the diversity and availability of temporal information in linked open data. In *The Semantic Web–ISWC 2012 : 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I 11*, pages 492–507. Springer, 2012.

[42] Simon Saunders. How relativity contradicts presentism. *Royal Institute of Philosophy Supplements*, 50 :277–292, 2002.

[43] Tom Stoneham. Time and truth : The presentism-eternalism debate. *Philosophy*, 84(2) :201–218, 2009.

[44] Fumiaki Toyoshima. Ontology of time for the digital humanities : A foundational view. In *Proceedings of the Joint Ontology Workshop JOWO 2019*, volume 2518 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[45] Octavian Udrea, Diego Reforgiato Recupero, and VS Subrahmanian. Annotated rdf. *ACM Transactions on Computational Logic (TOCL)*, 11(2) :1–41, 2010.

[46] Hsien-Tseng Wang and Abdullah Uz Tansel. Temporal extensions to rdf. *Journal of Web Engineering*, 2019.

[47] Christopher A. Welty and Richard Fikes. A reusable ontology for fluents in OWL. In Brandon Bennett and Christiane Fellbaum, editors, *Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, FOIS 2006, Baltimore, Maryland, USA*, volume 150 of *Frontiers in Artificial Intelligence and Applications*, pages 226–236. IOS Press, 2006.

[48] Fu Zhang, Ke Wang, Zhiyin Li, and Jingwei Cheng. Temporal data representation and querying based on rdf. *IEEE Access*, 7 :85000–85023, 2019.

[49] Antoine Zimmermann, Nuno Lopes, Axel Polleres, and Umberto Straccia. A general framework for representing, reasoning and querying with annotated semantic web data. *Journal of Web Semantics*, 11 :72–95, 2012.

Nouveaux réseaux neuronaux profonds pour l'alignement d'ontologies

S. Menad[1], W. Laddada[1], S. Abdeddaïm[1], LF. Soualmia[1]

[1] Univ. Rouen Normandie, LITIS UR4108, 76000, Rouen

{safaa.menad1,wissame.laddada,said.abdeddaim,soualfat}@univ-rouen.fr

Résumé

L'alignement d'ontologies s'appuie souvent sur des approches lexicales. Cependant, avec le développement des modèles de langage basés sur les transformeurs, il est désormais possible de comparer des textes en se basant sur le contexte plutôt que sur les caractéristiques lexicales. Dans ce travail, nous proposons de nouveaux modèles neuronaux siamois qui optimisent une fonction d'apprentissage contrastif auto-supervisé sur des articles de la littérature scientifique. Les résultats obtenus sur plusieurs benchmarks montrent que les modèles proposés permettent d'améliorer différentes tâches biomédicales. Ensuite, nous exploitons ces modèles dans la tâche d'alignement d'ontologies biomédicales.

Mots-clés

Modèles de langage, Transformeurs, Modèles neuronaux siamois, Apprentissage sans exemple, Textes biomédicaux, Ontologies biomédicales, Alignement d'ontologies.

Abstract

Ontology alignment often relies on lexical approaches. However, with the development of transformer-based language models, it is now possible to compare texts based on context rather than lexical characteristics. In this work, we propose new siamese neural models that optimize a self-supervised contrastive learning function using scientific literature articles. The results obtained from multiple benchmarks demonstrate that the proposed models improve various biomedical tasks. Moreover, we apply these models to the task of biomedical ontology alignment.

Keywords

Language Models, Transformers, Siamese Neural Networks, Zero-shot Learning, Biomedical Texts, Biomedical Ontologies, Ontology Alignment.

1 Introduction

L'alignement d'ontologie joue un rôle important dans l'intégration de connaissances. Il permet de faire correspondre des entités sémantiquement liées provenant de différentes ontologies. Les ontologies de domaine contiennent souvent un grand nombre de classes, ce qui non seulement pose des problèmes d'évolutivité, mais rend également plus difficile la distinction entre des classes ayant des noms et/ou des contextes similaires mais représentant des objets différents. Les solutions d'alignement ontologique existantes reposent généralement sur la correspondance lexicale comme base et la combinent avec la correspondance structurelle et la réparation d'alignements basée sur la logique.

Récemment, l'apprentissage automatique a été proposé comme une alternative aux méthodes de correspondance lexicale et/ou structurelle. Par exemple, DeepAlignment [11] utilise des plongements de mots pour représenter les classes et calcule la similarité de deux classes en fonction de la distance euclidienne de leurs vecteurs de mots. Néanmoins, ces méthodes adoptent des modèles de plongement de mots généralistes non contextuels tels que Word2Vec.

Les modèles de langage basés sur des transformeurs préentraînés tels que BERT [4] peuvent apprendre des plongements de texte contextuels robustes. Bien que ces modèles donnent de bons résultats dans de nombreuses tâches de traitement automatique du langage naturel (TAL), ils n'ont pas encore été suffisamment étudiés dans la tâche d'alignement ontologique et de mapping de concepts. Dans cet article, nous introduisons nos modèles transformeurs que nous avons développés dans la tâche d'alignement d'ontologies en présentant comment ces modèles pourraient être utilisés pour mapper sémantiquement des entités provenant de différentes ontologies biomédicales.

Cet article se structure comme suit : dans la section 2 nous citons des travaux existants dans le domaine de l'alignement d'ontologies. Les section 3 et 4 sont dédiées à la description des modèles de langage que nous proposons, BioS-Transformers et BioS-MiniLM, ainsi que la fonction objectif d'apprentissage contrastif. Les sections 5 et 6 décrivent les premiers résultats d'alignement d'ontologie avec nos modèles siamois sur deux ontologies biomédicales. Nous concluons et ouvrons des perspectives en section 6.

2 Travaux existants

Plusieurs ontologies de domaine ou d'application sont utilisées pour un même objectif. Cependant, des redondances ou des relations manquantes entre les concepts issus de différentes ontologies peuvent exister. Dans la littérature, l'alignement d'ontologies constitue une solution pour remédier à cette hétérogénéité et permettre une inter-opérabilité

sémantique entre les applications qui reposent sur plusieurs ontologies. L'alignement d'ontologies peut être défini comme une amélioration sémantique entre les concepts, les rôles et les instances de plusieurs ontologies pour une application donnée.
Dans [23], les auteurs ont défini un système distribué comme un système reliant deux ontologies. En fonction de cette définition, trois sémantiques d'un système distribué sont distinguées : une sémantique distribuée simple où la représentation des connaissances est interprétée dans un seul domaine, une sémantique distribuée intégrée où chaque représentation locale des connaissances est interprétée dans son propre domaine, et une sémantique distribuée contextuelle où le domaine d'interprétation n'est pas global.
Dans notre travail, nous souhaitons aligner deux ontologies issues d'un seul domaine (ontologies biomédicales) à l'aide de transformeurs pré-entraînés. De ce fait, la sémantique déployée est une sémantique distribuée simple.
L'alignement d'ontologies résulte d'une tâche importante de mise en correspondance (Ontology Mathcing - OM) où un alignement est défini pour identifier les similarités entre les ontologies. En ce qui concerne la classification des systèmes d'alignement présentée dans [20], un alignement peut reposer sur des similarités terminologiques (par exemple, des labels, des commentaires, des attributs, etc.), structurelles (description d'ontologie), extensionnelles (instances) ou sémantiques (interprétation et raisonnement logique). De plus, en raison du faible niveau d'expressivité sémantique de certaines ontologies, des ressources externes peuvent être exploitées dans les approches d'alignement. Par exemple, c'était le cas dans l'étude [13] pour l'alignement de l'ontologie SNOMED CT (Systematized Nomenclature of Medicine-Clinical Terms) avec BioTopLite2, une ontologie de haut niveau.
En ce qui concerne la mise en correspondance des éléments d'ontologies, une étude de la littérature approfondie est présentée dans [18] pour décrire les ressources externes et leur utilisation. Les auteurs distinguent quatre catégories d'approches d'alignement utilisant ces ressources : les requêtes factuelles, où les données stockées dans la ressource sont simplement récupérées ; les approches reposant sur la structure, où les éléments structurels de la ressource sont exploités ; les approches statistiques/neuronales (Fine-TOM [9], DAEOM [22]), où des approches statistiques ou d'apprentissage profond sont appliqués à la ressource ; et les approches orientées logique où le raisonnement est déployé sur la ressource externe. Par exemple, dans [2], des stratégies terminologiques et structurelles ainsi qu'une ressource externe ont été employées pour aligner des ontologies biomédicales.
Tout comme CIDER-LM [21], notre approche d'alignement dépend de similitudes terminologiques calculées avec des approches neuronales mais aussi une similitude contextuelle. Nous exploitons des ressources externes sur lesquelles un apprentissage profond est appliqué afin de propager un contexte de similitude entre les éléments (propriétés et classes) de deux ontologies biomédicales. La différence entre les deux approches réside dans le modèle de représentation utilisé. Dans [21] c'est le modèle S-BERT [19] qui est utilisé, tandis que dans notre travail, nous appliquons les modèles BioSTransformers que nous avons développés et que nous détaillons ci-après.

3 Modèles siamois

Les sentence-transformers sont des modèles de langage qui ont été développés pour la tâche de calcul d'un score de similarité entre deux phrases. Ils utilisent des transformeurs pour des tâches liées aux paires de phrases : calcul de similarité sémantique entre phrases, recherche d'informations, reformulation de phrases, etc.
Ces transformeurs reposent sur deux architectures : les cross-encodeurs qui traitent la concaténation de la paire et les modèles siamois bi-encodeurs qui encodent en vecteur chacun des éléments de la paire. Sentence-BERT [19] est un bi-encodeur basé sur BERT permettant de générer des plongements de phrases sémantiquement significatifs à utiliser dans des comparaisons de similarité textuelle.

4 Modèles de langage proposés

Les transformeurs siamois ont été initialement conçus pour transformer des phrases (de taille similaire) en vecteurs. Nous proposons dans notre approche de transformer dans le même espace vectoriel les termes MeSH, les titres et les résumés des articles PubMed en entraînant un modèle de transformeur siamois sur ces données. Nous voulons nous assurer qu'il y a une correspondance dans cet espace vectoriel entre le texte court et le texte long. Nous avons donc entraîné nos modèles avec des paires d'entrées (titre, terme MeSH) et (résumé, terme MeSH).
Sur ces données nous avons construit deux types de modèles [14] : le premier type est notre propre transformeur siamois (BioSTransformers) construit à partir d'un transformeur pré-entraîné sur des données biomédicales et le second est un transformeur siamois déjà pré-entraîné sur des données généralistes (BioS-MiniLM).

Dans cette étude, nous présentons une nouvelle variante de nos modèles BioSTransformers, appelée SBio_ClinicalBERT. Ce nouveau modèle a été pré-entraîné sur les notes cliniques de la base de données MIMIC. Ensuite, nous l'avons entraîné sur nos données biomédicales et utilisé pour résoudre la tâche d'alignement d'ontologies.
Le tableau 1 présente les résultats obtenus par ce modèle sur différents benchmarks selon le F1 score.
BioSTransformers. Pour construire les BioSTransformers, nous nous sommes inspirés du modèle Sentence-BERT [19] en remplaçant BERT par d'autres transformeurs entraînés sur des données biomédicales (bio-transformeurs). Nous avons sélectionné quatre bio-transformeurs BlueBERT [17], PubMed BERT [6], BioELECTRA [10] et BioClinicalBERT [1].
Pour l'entraînement, nous utilisons une fonction objectif d'apprentissage contrastif auto-supervisé basée sur la fonction de perte de classement négatif

multiple [8] dite MNRL (Multiple Negative Ranking Loss) dans le package Sentence-Transformers (https ://www.sbert.net/docs/package_reference/losses.html /#multiplenegativesrankingloss). La MNRL n'a besoin que des paires positives en entrée (le titre ou le résumé et un terme MeSH associé à l'article dans notre cas). Pour une paire positive (titre_i ou résumé_i, MeSH_i), la MNRL considère que chaque paire (titre_i ou résumé_i, MeSH_j) avec $i \neq j$ dans le même batch est négative.

Modèle/Corpus	HoC	PubMedQA	BioASQ
S-BioClinical	0.457	0.652	0.714

TABLE 1 – Évaluation de notre modèle sur les différents benchmarks selon le F1 score.

5 Alignement d'ontologies

Pour mieux comprendre notre cas d'utilisation, qui illustre un alignement d'ontologies biomédicales, nous présentons dans cette section quelques définitions inspirées d'autres travaux [18, 5, 16] et adaptées à notre objectif. La Figure 1 résume le processus d'un alignement d'ontologies suivant les définitions présentées dans cette section.

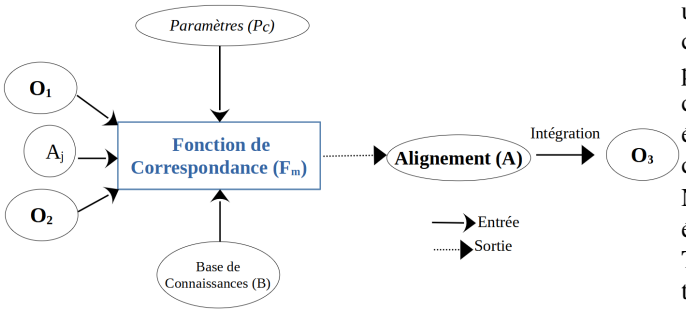


FIGURE 1 – Processus d'alignement d'ontologies (inspiré de [20]).

Définition d'ontologie : Nous considérons une ontologie O_i comme un ensemble de vocabulaire défini au moyen de taxonomies pour décrire un domaine d'application. Ce vocabulaire est considéré comme un ensemble d'éléments $e_i =< C_i, R_i, I_i >$; avec C_i pour décrire l'ensemble des concepts, R_i regroupe les relations pour relier les concepts et I_i compose l'ensemble des instances pour interpréter les concepts et les relier avec R_i. La sémantique d'une ontologie O_i peut être enrichie en définissant des axiomes (X_i) formalisés via la logique de description ou les logiques du premier ordre.

Alignement d'ontologies : un alignement décrit la correspondance entre deux ontologies. Étant donné deux ontologies O_1 et O_2, nous définissons l'alignement A comme un ensemble de triplets. Chaque triplet est spécifié par la terminologie de la relation binaire $r(e_1, e_2)$, où r représente la relation entre les deux éléments $e_1 \in O_1$ et $e_2 \in O_2$. L'alignement définit donc, le processus de recherche de ces ensembles de correspondance. Un score de confiance c peut également être ajouté au triplet de correspondance pour mesurer la similarité entre e_1 et e_2 (par exemple, la valeur de $c \in [0,1]$).

Système de correspondance : il peut être défini comme une fonction de correspondance ayant plusieurs paramètres pour calculer la similarité entre les entités. Soit $F_m(O_1, O_2, A_j, P_c, B)$ avec P_c comme étant le paramètre qui précise la valeur de confiance de similarité et B comme étant l'ensemble de ressources externes utilisées pour trouver (ou pas) un alignement A_j entre l'élément e_1 et e_2.

Intégration d'ontologies : en considérant le travail présenté dans [16], nous définissons l'intégration d'ontologie comme un enrichissement sémantique d'une ontologie cible O_1 en exploitant des éléments d'une ontologie source O_2. Le résultat est une nouvelle ontologie O_3 définie par l'alignement $A =< r_j, e_{1,j}, e_{2,j}, c_j >$.

6 Modèles pour l'alignement

Dans cette section, nous décrivons notre approche pour aligner les éléments de différentes ontologies biomédicales en utilisant nos modèles siamois décrits précédemment. Ainsi, ce dernier est un système central du processus de correspondance. Étant donné que les transformeurs fonctionnent comme des modèles de langage, il est important que les éléments de l'ontologie soient définis par des labels (ou des commentaires) et enrichis par des relations (propriétés).
Nous considérons le processus de correspondance comme étant un problème de similarité où notre modèle (BioS-Transformers) reçoit des éléments extraits à partir des ontologies en entrée et calcule leur similarité. En fonction du score de sortie, nous concluons si une correspondance est possible entre les deux éléments. Plusieurs ressources ont été utilisées et sont décrites ci-après :
I) *RxNorm* [15] est un standard nomenclature développé dans le domaine de traitement médical, par la NLM (*United States National Library of Medicine*- Bibliothèque américaine de médecine). La création de ce standard est motivée par le besoin d'unifier la terminologie qui permet de représenter les médicaments, mais également de permettre de l'inter-opérabilité sémantique. De plus, ce standard fournit une normalisation pour les médicaments cliniques et les noms de médicaments connexes. Ces derniers sont reliés à des vocabulaires couramment utilisés dans ce même domaine.
II) *ChEBI* [3] est un dictionnaire d'entités moléculaires décrivant les "petits" composants chimiques (182 374 classes, 10 relations). Les entités moléculaires sont soit des produits naturels, soit des produits synthétiques. ChEBI contient aussi des groupes (fait-partie d'entités moléculaires) et des classes d'entités. Ce dictionnaire comprend une classification ontologique, dans laquelle les relations entre les entités moléculaires ou les classes d'entités et leurs parents et/ou enfants sont spécifiées.

III) *DRON* [7] a été développée à partir de l'alignement d'entité de RxNorm et ChEBI. DRON est composée de 661 999 classes et 125 relations avec une profondeur de 27 niveaux.

IV) *DOID* [12] décrit des maladies et un vocabulaire médical via l'alignement de plusieurs ressources externes dans le but de lier les données biomédicales sur les gènes et les maladies. Elle est composée de 8 127 classes, 46 relations, avec une profondeur maximale de 13.

Notre cas d'utilisation décrit l'alignement d'éléments de deux ontologies biomédicales : DOID (*Human Disease Ontology* - ontologie des maladies humaines [1]) et DRON (*Drug Ontology* - ontologie des médicaments [2]. Le résultat de cet alignement représente une intégration d'ontologie dans laquelle chaque maladie est associée à une liste de médicaments potentiels. La finalité de ce processus d'alignement est d'intégrer l'ontologie résultante dans le système prédictif PrediBioOntoL. Afin de décrire la démarche du processus d'alignement, les phases listées dans [16] ont été adoptées.

6.1 La phase de prétraitement

Les données textuelles ont été extraites à partir des deux ontologies DOID et DRON via des requêtes SPARQL. Ces données concernent : (i) les classes (élément de DOID) qui décrivent une maladie [3]) et (ii) les métadonnées issues de ChEBI (Chemical Entities of Biological Interest - entités chimiques d'intérêt biologique) à partir desquelles l'ontologie DRON a été décrite. Ces métadonnées représentent des informations sur une maladie à travers une définition de propriété de données (métadonnées de ChEBI [4]). Cependant, aucune association n'est établie entre les ontologies DOID et DRON. Nous avons pu extraire un total de 13 678 maladies (DOID) et 3 295 métadonnées (DRON).

6.2 La phase de mise en correspondance

Le modèle BioSTransformers est utilisé comme fonction de correspondance, où les bases de connaissances externes représentent les données sur lesquelles le modèle est entraîné : PubMed dans un premier temps, puis sur MIMIC III (une base de données contenant les dossiers médicaux électroniques des patients). Pour cette étape, nous avons choisi le modèle SBio_ClinicalBERT. Par rapport à d'autres modèles, ce modèle fournit de bons résultats pour la comparaison des labels. Cela est dû au fait que ce modèle est entraîné sur les notes cliniques de MIMIC III.

6.3 Processus de mise en correspondance

Pour trouver les similarités entre les noms de maladies et les métadonnées, nous avons procédé de différentes manières. Dans un premier temps, nous avons pris seulement les noms de maladies à partir de l'ontologie DOID ($rdfs : label$) et avons calculé les similarités entre ces éléments et les métadonnées de l'ontologie DRON ($obo : IAO_0000115$).

1. https ://bioportal.bioontology.org/ontologies/DOID
2. https ://bioportal.bioontology.org/ontologies/DRON)
3. http ://purl.obolibrary.org/obo/
4. http ://purl.obolibrary.org/obo/$IAO_0000115$

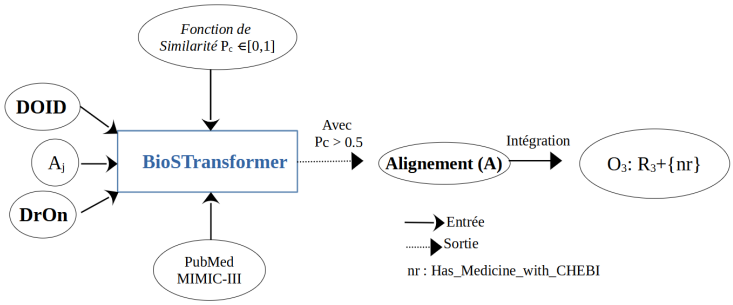


FIGURE 2 – Alignement de DOID et de DRON en utilisant BioSTransformers.

Nous avons ensuite amélioré notre processus en considérant deux approches qui prennent en compte d'autres éléments de DOID :

- La première consiste à *concaténer* plusieurs éléments de l'ontologie DOID. Ces éléments correspondent : au nom de la maladie ($rdf : label$), à sa définition ($obo : IAO_0000115$) et à plusieurs noms de maladies connexes ($oboInOwl : hasExactSynonym$). Nous appelons cette stratégie "multi-label". La concaténation est considérée comme une entrée pour BioSTransformers.
- La deuxième consiste à exploiter un seul élément à la fois à partir de DOID. Plus précisément, nous prenons en compte à chaque calcul d'une similarité soit le nom de la maladie ($rdf : label$) ou bien la définition de la maladie ($obo : IAO_0000115$) ou encore un seul nom de maladie connexe ($oboInOwl : hasExactSynonym$). Nous appelons cette stratégie "max-label". Ainsi, pour chaque élément de DRON pris en considération par BioSTransformers, la correspondance est établie avec un élément de DOID, en choisissant le score de similarité maximal résultant entre la métadonnée provenant de DRON ($obo : IAO_0000115$) et l'une des métadonnées de DOID ($rdf : label$ ou $oboInOwl : hasExactSynonym$ ou bien $obo : IAO_0000115$). Ce score doit être supérieur à 0,5.

6.4 La phase d'alignement

Les alignements générés sont des correspondances entre un seul concept de DOID et un seul concept de DRON (*alignement one-to-one*). Le type de correspondance est une *inclusion* entre les métadonnées qui définissent une classe ChEBI et celles qui définissent une maladie. Cet alignement est maintenu lorsque le score de confiance (le score de similarité) est supérieur au seuil de 0,5.

Si un alignement existe alors une nouvelle relation est définie entre la maladie et le concept ChEBI. Cette nouvelle relation permet de générer une troisième ontologie (ontologie d'intégration) enrichie par les ontologies DRON et DOID. Nous dénommons cette relation *Has_Medicine_with_CHEBI*. La Figure 2 illustre comment

les BioSTransformers sont utilisés dans la tâche d'alignement d'ontologie.
Le nombre d'alignements générés par les trois approches est illustré dans la Table 2. Nous constatons que la troisième approche produit le plus grand nombre d'alignements. Ainsi, le nom de la maladie n'est pas aussi représentatif que les autres métadonnées.
Les résultats obtenus sont très encourageants lors de l'utilisation de BioSTransformers pour trouver une similarité. Par exemple, dans DRON, l'élément "*CHEBI_27779*", qui compose le médicament sous le nom de "*Griseofulvin*", est défini par la métadonnée "*An oxaspiro compound produced by Penicillium griseofulvum. It is used by mouth as an antifungal drug for infections involving the scalp, hair, nails and skin that do not respond to topical treatment"*–"Un composé oxaspiro produit par Penicillium griseofulvum. Il est utilisé par voie orale comme médicament antifongique pour les infections du cuir chevelu, des cheveux, des ongles et de la peau qui ne répondent pas au traitement topique". Dans DOID, la maladie "*DOID_3136*" est définie par la métadonnée *"scalp dermatosis"*–"dermatose du cuir chevelu". Le processus de correspondance donne un score de similarité de 0,5608. Étant donné que le score de confiance est supérieur à 0,5 (seuil défini sans expérimentation), nous créons une nouvelle relation *"Has_Medicine_with_CHEBI(DOID_3136, CHEBI_27779)"*. Toutes les nouvelles relations peuvent être récupérées via une simple requête SPARQL.
La prochaine étape nécessaire à la validation des alignements consistera à les évaluer en utilisant des méthodes structurelles fondées sur les hiérarchies de concepts existantes dans chacune des deux ontologies utilisées. Une autre approche possible est l'utilisation du Metathesaurus de l'UMLS [5] (Unified Medical Language System) en tant que ressource d'évaluation.
Initialement, nous avons utilisé l'approche "multi-label". Ensuite, pour chaque maladie, nous avons cherché son médicament correspondant dans l'UMLS en utilisant son identifiant (CUI) et l'API UMLS [6], où le code est le CUI et nous avons récupéré son traitement par la relation sémantique "may_be_treated_by". Seules 490 maladies étaient associées à des codes CUI, tandis que pour les autres maladies, des codes alternatifs étaient nécessaires mais nous ne les avons pas utilisés ici.
Pour les 490 maladies ayant des codes CUI, nous avons cherché leurs médicaments correspondants dans l'UMLS et nous avons constaté que seuls 131 d'entre eux avaient la relation sémantique "may_be_treated_by" dans le réseau sémantique de l'UMLS. Nous avons également constaté dans l'UMLS que plusieurs maladies avaient le même médicament suggéré par nos modèles.
En analysant les résultats de manière plus approfondie et en examinant les définitions de chaque médicament, nous avons découvert que les incohérences étaient dues à notre modèle qui suggérait une entité chimique faisant partie de la composition du médicament. Cela est dû au fait que nous avons effectué l'alignement en utilisant DRON, qui est basé sur ChEBI, une ontologie d'entités chimiques.
En conclusion, nous pouvons affirmer que nos alignements sémantiques sont corrects même si le médicament suggéré ne correspond pas à celui fourni par l'UMLS. Cependant, notre méthode peut aider à enrichir l'UMLS avec des relations sémantiques supplémentaires qui n'y figurent pas encore.

Méthode	Nom de la maladie	multi-label	max-label
Nombre d'alignements	615	770	**1035**

TABLE 2 – Nombre d'alignements générés pour chaque mode de mise en correspondance.

7 Conclusion

Dans cette étude, nous avons proposé de nouveaux modèles siamois [14] BioSTransformers et BioS-MiniLM qui permettent d'améliorer différentes tâches biomédicales dans une configuration sans exemples (zéro shot). Ces modèles plongent des paires de textes dans le même espace de représentation et calculent la similarité sémantique entre des textes de différentes longueurs.
En outre, nous avons proposé d'exploiter nos modèles dans un scénario pratique qui consiste à aligner des entités de deux ontologies biomédicales distinctes afin d'établir de nouvelles relations.
L'approche a été instanciée dans un premier temps entre les deux ontologies DOID et DRON, dans le but de proposer un médicament potentiel pour une maladie donnée. Une évaluation des alignements obtenus est en cours et compte-tenu des premiers résultats, l'intégration d'autres ontologies (par exemple, les effets indésirables liés aux médicaments, ou d'autres ressources de médicaments comme DrugBank) est prévue. Enfin, la validation de l'approche proposée pour en démontrer sa généricité sur des ressources en Français est également envisagée.

Références

[1] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[2] Watson Wei Khong Chua and Jung jae Kim. Boat : Automatic alignment of biomedical ontologies using term informativeness and candidate selection. *Journal of Biomedical Informatics*, 45(2) :337–349, 2012.

[3] Kirill Degtyarenko, Paula Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcántara, Michael Darsow, Mickael Guedj, and Michael Ashburner. Chebi : A database and ontology

5. https ://www.nlm.nih.gov/research/umls/index.html
6. https ://uts-ws.nlm.nih.gov/rest/content/current/CUI/code/relations ?includeAdditionalRelationLabels=may_be_treated_by&apiKey

for chemical entities of biological interest. *Nucleic acids research*, 36 :D344–50, 02 2008.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[5] Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

[6] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1) :1–23, jan 2022.

[7] Josh Hanna, Eric Joseph, Mathias Brochhausen, and William Hogan. Building a drug ontology based on rxnorm and other sources. *Journal of biomedical semantics*, 4 :44, 12 2013.

[8] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv :1705.00652*, 2017.

[9] Sven Hertling, Jan Portisch, and Heiko Paulheim. Matching with transformers in melt. 09 2021.

[10] Kamal raj Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. BioELECTRA : Pretrained biomedical text encoder using discriminators. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 143–154, Online, June 2021. Association for Computational Linguistics.

[11] Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. Deepalignment : Unsupervised ontology matching with refined word vectors. In *Proceedings of NAACL-HLT, 787–798.*, pages 787–798, 2018.

[12] Schriml Lynn, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren Kibbe. Disease ontology : A backbone for disease semantic integration. *Nucleic acids research*, 40 :D940–6, 11 2011.

[13] Mélissa Mary, Lina Soualmia, Xavier Gansel, Stéfan Darmoni, Daniel Karlsson, and Stefan Schulz. Ontological representation of laboratory test observables : Challenges and perspectives in the snomed ct observable entity model adoption. pages 14–23, 05 2017.

[14] Safaa Menad, Saïd Abdeddaïm, and Lina Fatima Soualmia. Biostransformers : Modèles de langage pour l'apprentissage sans exemple dans des textes biomédicaux. In Catherine Faron and Sabine Loudcher, editors, *Extraction et Gestion des Connaissances, EGC 2023, Lyon, France, 16 - 20 janvier 2023*, volume E-39 of *RNTI*, pages 409–416. Editions RNTI, 2023.

[15] Stuart J Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs : RxNorm at 6 years. *Journal of the American Medical Informatics Association*, 18(4) :441–448, 04 2011.

[16] Inès Osman, Sadok Ben Yahia, and Gayo Diallo. Ontology integration : Approaches and challenging issues. *Information Fusion*, 71 :38–63, Jul 2021.

[17] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing : An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy, August 2019. Association for Computational Linguistics.

[18] Jan Portisch, Michael Hladik, and Heiko Paulheim. Background knowledge in ontology matching : A survey. *Semantic Web*, pages 1–55, 09 2022.

[19] Nils Reimers and Iryna Gurevych. Sentence-BERT : Sentence embeddings using Siamese BERT-networks. In *Proceedings of (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[20] Pavel Shvaiko and Jérôme Euzenat. Ontology matching : State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25 :158–176, 2013.

[21] Javier Vela and Jorge Gracia. Cross-lingual ontology matching with cider-lm : results for oaei 2022. 2022.

[22] Jifang Wu, Jianghua Lv, Haoming Guo, and Shilong Ma. Daeom : A deep attentional embedding approach for biomedical ontology matching. *Applied Sciences*, 10(21), 2020.

[23] Antoine Zimmermann and Jérôme Euzenat. Three semantics for distributed systems and their relations with alignment composition. In *The Semantic Web - ISWC 2006*, pages 16–29, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

Des données tabulaires aux graphes de connaissances : état de l'art des méthodes d'interprétation sémantique de tables

Jixiong Liu[1,2], Viet-Phi Huynh[1], Yoan Chabot[1], Raphaël Troncy[2]

[1] Orange, France
[2] EURECOM, Sophia Antipolis, France

yoan.chabot@orange.com

Résumé

Les données tabulaires sont omniprésentes sur le Web et dans les entrepôts de données des entreprises. Ces tableaux contiennent des informations pouvant potentiellement devenir des connaissances après une étape d'interprétation sémantique de tables se basant sur un graphe de connaissances. Ce papier propose un état de l'art des différentes tâches et méthodes existantes pour mener à bien cette interprétation. Dans un premier temps, nous proposons une nouvelle classification des tableaux reflétant la diversité et la complexité de ces structures. Nous décomposons ensuite le problème de l'interprétation sémantique en cinq sous-tâches et passons en revue trois familles d'approches au travers du prisme des corpus d'évaluation proposés par la communauté.

Mots-clés

Interprétation sémantique de tables, Annotation, Données tabulaires, Graphes de connaissances

Abstract

Tabular data are widely spread on the Web and in corporate data repositories. They contain information that can potentially become knowledge after a step called semantic interpretation based on a knowledge graph. This paper provides a state of the art of the different tasks and methods to carry out this interpretation. First, we propose a new classification of tabular data reflecting the diversity and complexity of these structures. We then decompose the problem of semantic interpretation of tables into five sub-tasks and review three families of approaches through the prism of evaluation corpora proposed within the community.

Keywords

Semantic Table Interpretation, Table annotation, Tabular data, Knowledge graph

1 Introduction

Les formats de données tels que CSV et XLS sont couramment utilisés en entrée d'algorithmes d'apprentissage automatique. Ainsi, l'interprétation des données tabulaires est une tâche ayant attiré beaucoup d'attention ces dernières années, avec notamment la cristallisation des efforts de recherche autour de challenges scientifiques comme SemTab [23]. Pour rendre les données tabulaires intelligibles, l'idée principale est de trouver des correspondances entre les éléments composant le tableau et les entités/concepts/relations décrits dans les graphes de connaissances (KG) encyclopédiques comme DBpedia [7] et Wikidata [53], ou spécifiques à l'entreprise. Ce problème est connu sous le nom d'annotation de données tabulaires (ou STI en anglais pour Semantic Table Interpretation). Les KG peuvent être utilisés pour guider l'interprétation sémantique tout en étant eux-mêmes les artefacts pouvant être enrichis ou corrigés par le résultat de l'interprétation. L'annotation des données tabulaires avec des entités sémantiques ouvre la voie à une utilisation plus intelligente des données. Elle permet d'envisager notamment des services basés sur ces nouvelles informations sémantique (e.g., indexation, recherche et recommandation d'informations à un niveau conceptuel) et de contribuer à l'amélioration des systèmes de questions/réponses.

L'interprétation automatique des données tabulaires est un problème complexe en raison du contexte limité disponible pour résoudre les ambiguïtés, du format de présentation des tableaux, et du caractère incomplet des KG vis à vis des connaissances présentes dans les tableaux. Ce papier vise à définir les différentes sous-tâches d'annotation de données tabulaires et à passer en revue les méthodes qui ont été proposées à ce jour, ainsi que leurs performances sur des ensembles de données d'évaluation bien établis dans la communauté.

La structure de ce papier est la suivante. La section 2 définit les notions essentielles inhérentes aux données tabulaires et propose une nouvelle taxonomie des types de tableaux. La section 3 définit ensuite les tâches liées à l'annotation de données tabulaires. La section 4, passe en revue les représentants de trois familles d'approches (non mutuellement exclusives) respectivement basées sur des heuristiques, sur de l'ingénierie de caractéristiques (ou feature engineering) et de l'apprentissage profond. La section 5 évalue les performances des systèmes STI puis la section 6 liste les défis scientifiques subsistants. Le lecteur peut se référer à [28] pour plus d'informations sur les jeux de données, les benchmarks ou encore les méthodes utilisées dans chacune des approches de l'état de l'art.

2 Données tabulaires

La première source d'information d'un système STI est la table elle-même (dans la suite du papier, les termes "table" et "tableau" seront utilisés de manière équivalente). Un tableau est un arrangement bidimensionnel de données comportant n lignes et m colonnes. Une cellule est l'élément de base d'un tableau où $\mathcal{T}_{ij}$ $(0 \leq i \leq n-1, 0 \leq j \leq m-1)$ indique la cellule de la ligne i et de la colonne j du tableau $\mathcal{T}$. Outre les données qu'elles contiennent, les métadonnées et le contexte dans lequel les tables apparaissent constituent des informations précieuses pour l'inteprétation. Par exemple, si un tableau a été publié sur une page web décrivant la Bundesliga, cette table est probablement plus en rapport avec le football qu'avec n'importe quel autre sport. Il est ainsi utile de collecter à la fois le tableau lui-même et ses métadonnées lors de l'extraction des données.

Avant d'interpréter un tableau, il est important d'identifier son type afin de prendre en compte ses spécificités dans le processus de STI. Étant donné l'importante hétérogénéité des tables en termes de format, de provenance et d'utilisation, nous introduisons dans cette section une classification des types de tableaux (figure 1) basée sur les classifications existantes avec une analyse plus approfondie des tables relationnelles. Cette classification des tables vise à faciliter la définition du champ d'application des approches et à aider à mieux décrire les défis liés aux tâches de STI.

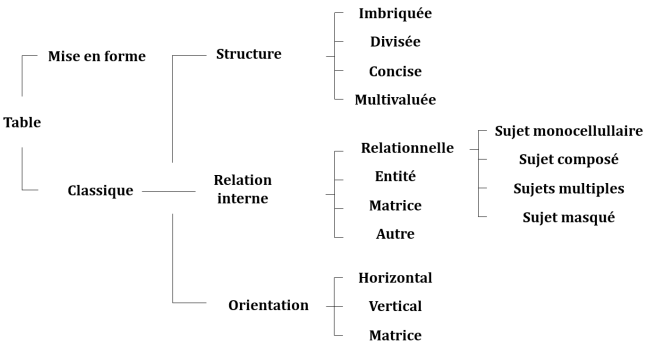


FIGURE 1 – Classification des types de tables.

Les tableaux peuvent être divisés en deux grandes catégories. **Les tables de mise en forme** sont utilisées pour structurer et formater les pages Web. Ces tables ne contiennent pas de relations sémantiques et sont utilisées pour organiser visuellement le contenu d'une page afin de maximiser le confort de l'utilisateur et l'ergonomie d'un site.

Les tables dites classiques contiennent des connaissances. Ces tableaux présentent un niveau élevé de cohérence (syntaxique et sémantique) entre les lignes et les colonnes. Les tableaux de cette classe contiennent des connaissances qui peuvent être interprétées et constituent donc des données d'entrée pour le processus de STI.

Nous proposons ensuite de classer les tableaux classiques en fonction de trois dimensions non mutuellement exclusives : la structure, les relations internes et l'orientation. Les types de tableaux sont ensuite formés par la composition de ces dimensions. Par exemple, le tableau représenté dans la figure 2(b) sur les lignes de chemin de fer est un tableau concis (dimension structurelle), un tableau horizontal (orientation) et un tableau relationnel à sujet composé (relation interne). Dans la suite, nous définissons plus en détail chacune de ces trois dimensions.

La sous-classe **structure** de notre classification est divisée en quatre types de tableaux. Les **tableaux imbriqués** contiennent un ou plusieurs tableaux dans une ou plusieurs de leurs cellules. Les **tableaux divisés** sont des tableaux pouvant être divisés en sous-tableaux. Les **tableaux concis** contiennent des cellules fusionnées afin d'éviter les répétitions de cellules faisant référence au même contenu dans les lignes et/ou les colonnes. Les **tableaux multivalués** contiennent plusieurs valeurs (sous la forme d'une énumération non structurée par exemple) dans une seule cellule.

La sous-classe **relation interne** prend en compte les relations sémantiques entre les cellules. Dans notre classification, nous proposons les types suivants. Les **tableaux relationnels** sont des structures dans lesquelles chaque ligne (ou colonne) fournit des informations sur une entité spécifique, et les colonnes (ou lignes) correspondantes représentent des attributs qui décrivent l'entité. Les **tables d'entité**, également appelées tables attribut-valeur, sont utilisées pour décrire une entité unique. Une table d'entité énumère les attributs de l'entité (e.g. infobox Wikipédia). Les **matrices** présentent un arrangement bidimensionnel de données qui doivent être lues simultanément horizontalement et verticalement. Une matrice associe des paires $(\text{row}, \text{column})$ aux valeurs des cellules par le biais d'une propriété unique. Les **autres** tables contiennent des informations mais ne correspondent pas aux types susmentionnés (e.g. les énumérations et les calendriers).

La littérature considère les tables relationnelles comme une feuille dans les taxonomies de types de tables [26]. Cependant, les tables relationnelles présentent une diversité importante, notamment dans la représentation des entités. Nous proposons de les classer plus finement en fonction des caractéristiques de leurs sujets. Le **sujet** d'une ligne d'un tableau relationnel horizontal (resp. d'une colonne d'un tableau relationnel vertical) est une entité qui est décrite par les ensembles de cellules dans cette ligne (resp. colonne). Nous introduisons quatre sous-types de tableaux relationnels (figure 2).

Les **tableaux à sujet monocellulaire** associent chaque ligne d'un tableau horizontal (ou chaque colonne d'un tableau vertical) à un seul sujet. Les mentions (i.e. label représentant une entité) des sujets sont indiquées dans une seule colonne (resp. ligne). Par exemple, dans la figure 2(a), la colonne "Department" contient les sujets. Les autres colonnes décrivent les sujets. Les **tableaux à sujet composé** nécessitent la combinaison de plusieurs cellules pour former le sujet de chaque ligne (resp. colonne). Par exemple, dans le tableau de la figure 2(b), on peut identifier les sujets (classes de train particulières) en fusionnant les colonnes "Lines", "Manufacturer" et "Class". Les **tableaux à sujets multiples** contiennent des cellules qui se réfèrent à des sujets différents tout en étant dans la même ligne.

(a)

Department	Area (km²)	Population (2011)[37]	Municipalities
Paris (75)	105.4	2 249 975	1 (Paris)
Hauts-de-Seine (92)	176	1 581 628	36 (list)
Seine-Saint-Denis (93)	236	1 529 928	40 (list)
Val-de-Marne (94)	245	1 333 702	47 (list)
Petite Couronne	657	4 445 258	123
Paris + Petite Couronne	762.4	6 695 233	124

(b)

Lines	Manufacturer	Class	Image	Number	Car numbers	Built
BART main lines	Rohr	A		59	1164–1276	1968–1975
	Rohr	B		380	1501–1913	1971–1975
	Alstom	C1		150	301–450	1987–1989
	Morrison-Knudsen	C2		80	2501–2580	1994–1996[67]
	Bombardier	D		*310*	3001–3310	2012–
	Bombardier	E		*465*	4001–4465	2012–
Oakland Airport Connector	DCC Doppelmayr	Cable Liner		4	1.3–4.3	2013
eBART	Stadler	GTW		8	101–108	2014-2018

(c)

Release year	Album	Artist/s	Nationality	Worldwide sales (in millions)	Ref(s)
2002	*Come Away With Me*	Norah Jones	United States	**23.9**	[3]
2000	*The Marshall Mathers LP*	Eminem	United States	**23.29**	[4]
2002	*The Eminem Show*	Eminem	United States	**22.95**	[5]
2000	*Hybrid Theory*	Linkin Park	United States	**20.8**	[6]
2015	*25*	Adele	United Kingdom	**20.41**	[7]

(d)

3	11 juin 1998	Italie	2 - 2	Chili
4	11 juin 1998	Cameroun	1 - 1	Autriche
19	17 juin 1998	Chili	1 - 1	Autriche
20	17 juin 1998	Italie	3 - 0	Cameroun
33	23 juin 1998	Italie	2 - 1	Autriche
34	23 juin 1998	Chili	1 - 1	Cameroun

FIGURE 2 – (a) Table à sujet monocellulaire [a], (b) Table à sujet composé [b], (c) Table à sujets multiples [c], (d) Table à sujet masqué [d].

a. https://en.wikipedia.org/wiki/France#Major%20cities
b. https://en.wikipedia.org/wiki/Bay_Area_Rapid_Transit
c. https://en.wikipedia.org/wiki/List_of_best-selling_albums_of_the_21st_century
d. https://fr.wikipedia.org/wiki/Coupe_du_monde_de_football_1998

Dans la figure 2(c), une ligne est composée de deux sujets : "Artist(s)" est le sujet de la colonne "Nationality", tandis que "Album" est le sujet des colonnes "Release year", "Artist(s)", "Worldwide sales" et "Ref(s)". Les **tableaux à sujet masqué** ne mentionnent pas explicitement le sujet de chaque ligne (resp. colonne). Par exemple, dans la figure 2(d), chaque ligne décrit le résultat d'un match de football, mais la mention du match lui-même n'est pas explicite dans le tableau.

La sous-classe **orientation** spécifie la direction des relations. Connaître le sens des relations structurant un tableau simplifie son interprétation en permettant d'associer les bons attributs à un sujet donné par exemple. Dans les **tableaux horizontaux**, les sujets sont décrits horizontalement, ce qui signifie que chaque ligne décrit un sujet différent. Dans les **tableaux verticaux**, les sujets sont décrits verticalement, ce qui signifie que chaque colonne décrit un sujet différent. Les **matrices**, quant à elles, doivent être interprétées cellule par cellule (et non ligne par ligne ou colonne par colonne) tout en tenant compte des en-têtes horizontaux et verticaux.

3 Interprétation automatique de tables

Après avoir donné les définitions nécessaires à l'étude du domaine, cette section présente le processus de STI en détaillant les cinq tâches qui le composent.

Une tâche d'annotation peut être définie par les éléments du tableau qui doivent être annotés et par le type de candidats considérés (individus, concepts ou propriétés du KG).

Nous proposons de décomposer le domaine de l'interprétation de tables en cinq tâches principales illustrées dans la figure 3 : "cell-entity annotation", "column-type annotation", "columns-property annotation" [23], la thématisation ("topic annotation"), et la tâche de correspondance ligne-instance [41].

L'annotation de cellules avec des entités (CEA) a pour but d'annoter une cellule avec une entité d'un KG. Par exemple, dans la figure 3, la tâche de CEA permet de faire correspondre la mention "Suisse" avec l'entité Wikidata Q165141. **L'annotation de colonnes avec des types** (CTA) a pour objectif de faire correspondre une colonne avec un type sémantique (classe) du KG. La difficulté de la tâche CTA réside dans la sélection du type le plus pertinent. Une entité peut être associé à plusieurs types représentés dans des arbres hiérarchiques complexes (par exemple, la topologie des types de Wikidata). Le type sélectionné pour une colonne donnée doit être représentatif des individus qu'elle contient et véhiculer un maximum d'informations. Si le type choisi est trop général (par exemple, la deuxième colonne du tableau de la figure 3 est annotée comme une "geographic entity" (Q27096213) plutôt que comme une "city of Switzerland" (Q1545591)), l'annotation portera peu d'informations. Inversement, un type trop spécifique peut être moins représentatif des valeurs d'une colonne. **L'annotation de colonnes avec des propriétés** (CPA) vise à annoter une paire de colonnes avec une propriété du KG. Par exemple, la relation entre la dernière colonne et la colonne entourée en orange dans la figure 3 doit correspondre au prédicat "number of points/goals/set scored" (P1351). **La thématisation** vise à annoter l'en-

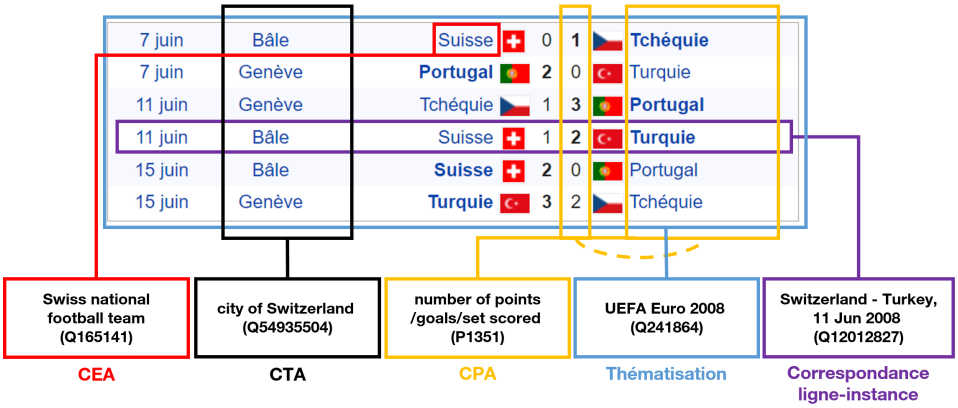


FIGURE 3 – Illustration des cinq tâches de STI sur une table décrivant les résultats du groupe A de l'Euro UEFA 2008 [a].

a. https://fr.wikipedia.org/wiki/Championnat_d%27Europe_de_football_2008#1er_tour_-_phase_de_groupes

semble du tableau avec un concept ou une entité du KG cible. La figure 3 révèle que le tableau entier est lié à l'entité "UEFA Euro 2008" (Q241864) dans Wikidata. **La correspondance ligne-instance** annote une ligne entière d'une table relationnelle avec une entité du KG. Dans cette tâche, chaque ligne est traitée comme une entité, qui est considérée comme le sujet de la ligne. La tâche de correspondance ligne-instance diffère de la tâche CEA car elle peut permettre de découvrir davantage d'entités en s'appuyant sur le contexte de la ligne, en particulier dans le cas où le sujet de la ligne est caché. Par exemple, dans la figure 3, la quatrième ligne est représentée par ("Switzerland - Turkey, 11 Jun 2008" (Q12012827)) qui ne peut pas être identifiée par le CEA. L'ensemble de ces tâches permettent d'établir des correspondances entre les tableaux et le graphe de connaissances. Cette étape constitue une première étape pour la génération de triplets à partir du contenu du tableau.

Enfin, les tâches de STI utilisent les KG comme sources d'information et comme références pour la production d'annotations. La colonne KG de la Table 1 fournit les KG qui ont été utilisés dans chaque système STI examiné dans ce papier. Les KG sont des éléments essentiels pour soutenir le processus de STI. En effet, comprendre le contenu d'un tableau revient à identifier les entités mentionnées dans les cellules du tableau et les relations entre elles. En fonction de leur contenu, les KG peuvent être classés en KG spécifiques à un domaine, en KG encyclopédiques ou en KG de sens commun. Les KG les plus couramment utilisés pour l'interprétation des tableaux sont **DBpedia** [7], **Wikidata** [53], **Freebase** [8] et **YAGO** (Yet Another Great Ontology) [50]. Les KG spécifiques à un domaine sont très peu utilisés à ce jour pour les tâches de STI comme en témoigne la table 1.

4 Approches

Dans cette section, nous passons en revue les approches de STI. Plusieurs jeux de données ont été proposés pour évaluer les approches de STI. Certains d'entre eux sont des références pour l'évaluation dans lesquels les composants des tableaux (cellules, lignes, colonnes ou paires de cellules) sont associés à des éléments de KG (entité, classe ou propriété), tandis que d'autres jeux de données proposent des tableaux de qualité pour permettre l'entrainement de système. Plusieurs jeux de données sont utilisés pour comparer les approches de STI dans cet état de l'art : Limaye [27], T2D [43], WDC [26], TabEL [5], Zhang et al. [59] et SemTab 2019, 2020, 2021 [1]. Parmi les cinq tâches présentées dans la section précédente, la littérature se concentre principalement sur le CTA, le CEA et le CPA. Nous proposons de classer les systèmes STI selon trois familles représentatives de leur méthodologie intrinsèque : les méthodes heuristiques (section 4.1), les méthodes basées sur l'ingénierie de caractéristiques (section 4.2) et les méthodes basées sur l'apprentissage profond (section 4.3). La table 1 présente plus de détails sur cette classification, notamment les algorithmes représentatifs, les tâches ciblées, les éléments de tableau utilisés, les KG cibles et l'année de publication.

4.1 Approches heuristiques

La famille des approches heuristiques regroupe diverses approches de STI utilisant des algorithmes simples et ne nécessitant pas d'efforts significatifs d'ingénierie de caractéristiques ou d'apprentissage. En effet, les tâches de STI sont ici effectuées à l'aide de techniques heuristiques telles que les mesures de similarité de chaînes de caractères [31, 39, 55], le vote majoritaire [61], TF-IDF [39, 51] ou les méthodes probabilistes [35]. Le contexte du tableau, y compris l'en-tête, le titre et les cellules voisines [22, 55] peuvent être pris en compte par ce type d'approches mais pas systématiquement. Nous identifions deux sous classes d'approches heuristiques. Tout d'abord, les approches basées sur les opérations de lookup travaillent avec un ensemble initial d'entités candidates déterminé par un service de recherche. Après avoir généré des candidats, ces méthodes les classent à l'aide de différentes métriques reposant sur les éléments du tableau (par exemple, les cellules,

1. http://www.cs.ox.ac.uk/isg/challenges/sem-tab/

TABLE 1 – Les approches de STI sont classifiées en trois familles. "R2I" correspond à la tâche de correspondance ligne-instance ; "TA" correspond à la thématisation ; "$\mathcal{T}_{i*}$" indique que des informations de la ligne sont utilisées pour annoter une cellule donnée ; "$\mathcal{T}_{*j}$" indique que l'approche tire parti des informations portées par la colonne étudiée (CEA, CTA) ou les colonnes voisines (CPA) ; "$\mathcal{T}_{0*}$" signifie que l'approche utilise les informations contenues dans l'en-tête ; "$\mathcal{T}_{**}$" indique que l'approche entraine un modèle sur l'ensemble des éléments contenus dans la table incluant les influences entre colonnes ; "$\mathcal{T}_{out}$" indique que l'approche utilise, en complément de la table, des métadonnées et le contexte associés à la table.

Approches			Tâches					Elements utilisés					KG	Source	Année de publication
Famille		Algorithme	CEA	CTA	CPA	R2I	TA	$\mathcal{T}_{i*}$	$\mathcal{T}_{*j}$	$\mathcal{T}_{0*}$	$\mathcal{T}_{**}$	$\mathcal{T}_{out}$			
Heuristique	Lookup	Venetis et al. [52]		√	√				√				Ad-hoc	Web Tables	2011
		Wang et al. [55]		√				√	√	√			Probase	Wikipedia Tables	2012
		Deng et al. [15]		√					√				FreeBase, YAGO	Wikipedia Tables	2013
		Sekhavat et al. [44]			√				√				DBpedia	Web Tables	2014
		TabEL [5]	√					√	√				YAGO	Limaye	2015
		ADOG [39]	√	√	√			√	√				DBpedia	SemTab 2019	2019
		Tabularisi [51]	√	√	√				√				DBpedia	T2D, VizNet	2019
		C^2 [25]		√				√	√	√			DBpedia, Wikidata	Limaye, ISWC2017, SemTab 2019, T2D	2020
		Magic [47]	√	√	√				√	√			DBpedia, Wikidata	SemTab 2021	2021
		Alobaid et al. [2]		√						√			DBpedia	SemTab 2021, T2D	2022
	Itérative	Zwicklbauer et al. [61]		√					√				DBpedia	Wikipedia Tables	2013
		T2K [42]			√	√	√	√	√				DBpedia	T2D	2015
		TableMiner+ [59]			√	√	√	√	√	√		√	Freebase	Limaye, IMDB, MusicBrainz	2017
		LOD4ALL [31]	√	√	√			√	√				DBpedia	SemTab 2019	2019
		CSV2KG [48]	√	√	√			√	√	√			DBpedia	SemTab 2019	2019
		MTab [35, 37, 38]	√	√	√			√	√	√			DBpedia, Wikidata	SemTab 2019-2021	2019
		LinkingPark [12]	√	√	√			√	√				Wikidata	SemTab 2020	2019
		DAGOBAH SL [20, 21, 22]	√	√	√			√	√				DBpedia,Wikidata	SemTab 2019-2022	2019
		MantisTable [14, 13]	√	√	√	√		√	√				DBpedia, Wikidata	SemTab 2019-2021	2019
		JenTab [1]	√	√	√			√	√				DBpedia, Wikidata	SemTab 2020-2021	2020
Ingénierie de caractéristiques		Limaye et al. [27]	√	√	√			√	√	√			YAGO	Limaye	2010
		Mulwad et al. [33, 32]	√	√	√			√	√	√			Wikitology	Limaye	2010
		SemanticTyper [40]		√					√				DBpedia	Museum	2015
		DSL [30]		√					√				DBpedia	City, Museum, Weather, Custom Soccer	2016
		Neumaier et al. [34]		√					√				DBpedia	Portail de données gouvernementales	2016
		NUMER [24]		√					√		√		DBpedia	NumDB	2018
Apprentissage profond	Modélisation du KG	Vasilis et al. [17]	√					√	√				Wikidata	Limaye, T2D, Wikipedia	2017
		Biswas et al. [6]					√					√	DBpedia	Wikipedia infobox	2018
		DAGOBAH Embeddings [10]	√	√				√	√	√			DBpedia, Wikidata	SemTab 2019	2019
		Radar Station [29]	√						√				Wikidata	Limaye, T2Dv2, SemTab 2020	2022
	Modélisation de la table	Sherlock [19]		√					√				DBpedia	T2D, VizNet	2019
		Sato [57]		√					√		√		DBpedia	VizNet	2019
		ColNet [11]		√				√	√				DBpedia	Limaye, T2Dv2	2019
		Guo et al. [18]		√					√			√	DBpedia	T2Dv2	2020
		Zhang et al. [58]	√		√			√	√				DBpedia	T2Dv2	2020
		TURL [16]	√	√	√			√	√	√	√	√	DBpedia	WikiGS, WikiTable, T2D	2020
		TCN [54]		√	√			√	√		√	√	-	Web Tables, WikiTable [16]	2021
		DUDUO [49]		√	√			√	√		√		-	WikiTable, VizNet	2021
		Singh et al. [46]			√				√	√		√	DBpedia	T2Dv2	2021
		Zhou et al [60]		√					√		√		DBpedia	Wikipedia Tables	2021

le type de colonnes, etc.). **Venetis et al.** [52] et **TabEL** [5] sont deux approches notables de cette sous-classe. Deuxièmement, les approches itératives sont construites à partir d'un système à base de lookup, avec une étape supplémentaire de désambiguïsation pour reclasser les entités candidates. Les techniques de désambiguïsation itératives jouent un rôle important dans l'amélioration des performances et de nombreuses approches performantes de l'état de l'art appartiennent à cette sous-classe, notamment **T2K** [42], **MTab** [35], **LinkingPark** [12], **JenTab** [1] et **DAGOBAH SL** [20, 21, 22].

4.2 Approches basées sur l'ingénierie de caractéristiques

Cette famille de méthodes extrait des caractéristiques statistiques et lexicales (telles que la distribution des valeurs numériques, l'occurrence des mentions de cellules, la similarité textuelle, etc.) des lignes et des colonnes du tableau et les utilise dans des modèles d'apprentissage automatique. Les algorithmes utilisés par cette famille sont, par exemple, SVM [33], Random Forest [30] et K-Nearest Neighbor [34]. La quantité et la qualité des données d'apprentissage, et par conséquent la qualité des caractéristiques d'entrée, ont un impact significatif sur la performance des modèles, comme indiqué dans [30]. En outre, nous observons que les méthodes à base d'apprentissage ciblent la tâche CTA en particulier, car les colonnes peuvent fournir plus de caractéristiques statistiques que d'autres cibles d'annotation. **Limaye et al.** [27] et **Mulwad et al.** [33] sont deux approches importantes de cette famille.

4.3 Approches basées sur l'apprentissage profond

L'apprentissage profond a connu un grand succès dans plusieurs domaines grâce à la disponibilité d'énormes quantités de données et de puissantes ressources informatiques. Il a attiré de plus en plus l'attention de la communauté du STI au cours des dernières années. Nous identifions deux courants principaux dans cette famille d'approches. Premièrement, la modélisation de KG se concentre sur l'apprentissage de plongement des entités représentant les cellules des tables (et non les cellules elles-mêmes). Plus précisément, les techniques de plongements de KG (par exemple, TransE [9] et TransH [56]) sont utilisées pour plonger les entités et leurs relations dans un espace vectoriel. Les modèles de STI reposent sur l'intuition que les entités d'une même colonne doivent présenter des similitudes sémantiques. Elles doivent donc être proches les unes

des autres dans l'espace de plongement au regard de la distance de similarité cosinus [17] ou de la distance euclidienne [10]. **DAGOBAH Embeddings** [10] et le module **Radar Station** [29] sont deux approches utilisant la modélisation de KG. Deuxièmement, la modélisation des tableaux considère directement le contenu textuel du tableau ainsi que les interactions intra-table et inter-table. La représentation des éléments de base du tableau comme les cellules ou les colonnes est apprise à l'aide de réseaux de neurones profonds [11, 19] ou de modèles de langage comme BERT [16, 49, 60].

5 Evaluation

Dans cette section, nous analysons plus en détail les performances, les forces et les faiblesses de chaque famille d'approches de STI. La Table 2 résume les performances des trois meilleurs systèmes avec les scores F1, AP, ou AF1 les plus élevés pour les tâches CEA, CTA et CPA sur les ensembles de données couramment utilisés par la communauté. Les scores AP et AF1 sont utilisés pour la tâche de CTA afin de prendre en compte la multiplicité des annotations possibles, avec des types plus ou moins génériques/plus ou moins porteurs d'informations. Les ensembles de données, la méthode de collecte des résultats et les métriques d'évaluation sont présentés plus en détail dans [28].

Sur la base de notre classification des approches de STI, nous observons que les systèmes heuristiques apparaissent dans les trois premiers systèmes pour tous les ensembles de données et toutes les tâches. En particulier, aucun des systèmes à base d'ingénierie de caractéristiques ou des systèmes basés sur l'apprentissage profond n'a atteint le podium pour les tâches d'appariement d'entités (CEA et correspondance ligne-instance). L'une des principales raisons est que, contrairement aux tâches de CTA ou CPA, qui valorisent des caractéristiques sur les colonnes ou des paires de colonnes, les caractéristiques qui peuvent être utilisées pour annoter des cellules ou des lignes sont plus rares. Cela limite donc les performances de ces systèmes.

Méthodes d'appariement VS méthodes d'apprentissage. Nous avons observé que les approches de STI reposent soit sur l'appariement (association d'une entité du KG et d'une cellule de la table) soit sur de l'apprentissage. L'appariement est la base des approches heuristiques, tandis que l'ingénierie de caractéristiques et les méthodes basées sur l'apprentissage profond reposent sur l'apprentissage de la représentation du tableau d'entrée. Ces approches peuvent également être combinées : les annotations apprises par les réseaux de neurones sont affinées à l'aide de techniques d'appariement utilisées lors d'un post-traitement (DAGOBAH Embeddings [10] et ColNet [11] en sont deux exemples). D'après nos observations, l'efficacité des opérations d'appariement dépendent fortement de la compatibilité entre la table et le KG cible. Par conséquent, ce type de technique souffre de l'incomplétude du tableau et des problématiques de knowledge shifting du KG. Les méthodes d'appariement sont moins résistantes au bruit que les méthodes d'apprentissage. De leur côté, les méthodes d'apprentissage nécessitent de grands ensembles de données d'entraînement qui ne sont pas toujours simples à collecter ou à générer. Certaines approches d'apprentissage limitent toutefois le nombre de candidats cibles pour pallier au manque de données d'apprentissage. La taille des tableaux constitue un autre défi pour les méthodes d'apprentissage. Certaines méthodes s'appuient sur les caractéristiques statistiques calculées à partir du tableau (par exemple, la longueur des mentions). Ces caractéristiques ne sont pas statistiquement stables si le nombre de cellules du tableau est faible.

L'essor de l'apprentissage profond. A partir de 2017, l'apprentissage profond a fait son entrée dans le domaine du STI. Par rapport aux approches d'ingénierie de caractéristiques, les réseaux de neurones profonds permettent au système de traiter les caractéristiques des tables plus efficacement, car l'étape d'ingénierie de caractéristiques est parfois difficile et longue à maintenir. Par exemple, Sherlock [19] est basé sur 1588 caractéristiques issues de colonnes. Pour remédier à ce problème, un apprentissage de bout en bout est préférable et de plus en plus utilisé, par exemple, la modélisation de KG à l'aide de plongements de graphes [17] et la modélisation de tables avec des modèles de type BERT[54]. Cependant, nous observons que les approches de modélisation de tables utilisant des modèles de langage ciblent toujours des tâches d'annotation de classes (CTA) ou de relations (CPA). La tâche d'annotation d'entités (CEA) n'a pas encore fait l'objet de travaux spécifiques de ce type, excepté TURL [16] proposant une matrice de visibilité pour décrire les connexions entre les éléments du tableau (par exemple, les cellules dans les mêmes colonnes, les cellules dans les mêmes lignes, etc.). En outre, de nombreux systèmes [49, 54, 60] simplifient la représentation des tableaux en ignorant l'ordre des lignes et des colonnes notamment.

Compromis entre l'efficacité et la précision. Les systèmes d'annotation sont généralement confrontés à un compromis entre efficacité et précision. TableMiner+ [59] introduit un appariement partiel dans lequel le calcul du CTA repose sur seulement huit lignes du tableau afin d'améliorer les performances. Cette stratégie rend en effet les systèmes plus rapides mais dégrade la précision. Par exemple, si l'on considère l'annotation d'une colonne contenant ["Joe Biden", "Donald Trump", "Barack Obama", "Abe Shinzo"], l'application de la correspondance partielle sur les trois premières cellules de la colonne produira "Présidents américains" comme type de cette colonne, alors que la réponse correcte est plus probablement "politiciens" puisque "Abe Shinzo" n'est pas un président américain mais un premier ministre japonais. Enfin, les systèmes dont le pipeline d'annotation comprend une étape de génération de candidats dépendront fortement du service de lookup d'entités utilisé. Cependant, les points d'accès publics imposent plusieurs limites à leur utilisation et l'obtention d'un ensemble de candidats avec une couverture souhaitable de la table cible peut prendre davantage de temps.

TABLE 2 – Top 3 des approches pour chaque jeu de données au regard du F1-score.

Jeux de données		CEA / Correspondance ligne-instance			CTA [a] / Thématisation			CPA		
Limaye		TabEL	TabEAno [36]	T2K ++	T2K ++	Guo et al	MantisTable	Mulwad et al.	T2K ++	TableMiner+
		0.894	0.88	0.87	0.88	0.852	0.84	0.89	0.80	0.76
T2D		TabEAno	Zhang et al.	Kruit et al.	ColNet	Alobaid et al. [2]	MantisTable	T2K ++	Singh et al.	MantisTable
		0.91	0.90	0.89	0.976	0.96	0.95	0.91	0.71	0.51
SemTab 2019	R2	MTab	CSV2KG	Tabularisi	MTab	CSV2KG	Tabularisi	CSV2KG	IDLab	Tabularisi
		0.911	0.883	0.826	1.414	1.376	1.099	0.881	0.877	0.790
	R3	MTab	CSV2KG	ADOG	MTab	CSV2KG	Tabularisi	MTab	CSV2KG	Tabularisi
		0.970	0.962	0.912	1.956	1.864	1.702	0.844	0.841	0.827
	R4	MTab	MantisTable	CSV2KG	MTab	CSV2KG	Tabularisi	MTab	CSV2KG	Tabularisi
		0.983	0.973	0.907	2.012	1.846	1.716	0.832	0.830	0.823
SemTab 2020	R1	MTab	LinkingPark	MantisTable	JenTab	LinkingPark	MTab	MTab	LinkingPark	JenTab
		0.987	0.987	0.982	0.962	0.926	0.885	0.971	0.967	0.963
	R2	MTab	DAGOBAH	LinkingPark	LinkingPark	MTab	DAGOBAH	MTab	LinkingPark	DAGOBAH
		0.995	0.993	0.993	0.984	0.984	0.983	0.997	0.993	0.992
	R3	MTab	LinkingPark	DAGOBAH	LinkingPark	MTab	DAGOBAH	MTab	DAGOBAH	bbw [45]
		0.991	0.986	0.985	0.978	0.976	0.974	0.995	0.993	0.989
	R4	MTab	LinkingPark	DAGOBAH	MTab	bbw	DAGOBAH	MTab	bbw	DAGOBAH
		0.993	0.985	0.984	0.981	0.98	0.972	0.997	0.995	0.995
	2T	MTab	bbw	DAGOBAH	DAGOBAH	MTab	LinkingPark	-	-	-
		0.907	0.863	0.830	0.743	0.728	0.686	-	-	-
SemTab 2021	R1 (DBpedia)	DAGOBAH	GBMTab	JenTab	JenTab	DAGOBAH	Magic	-	-	-
		0.945	0.692	0.607	0.46	0.422	0.159	-	-	-
	R1 (WikiData)	DAGOBAH	MTab	AMALGAM [3]	DAGOBAH	MTab	JenTab	-	-	-
		0.923	0.907	0.658	0.832	0.728	0.697	-	-	-
	R2-Hard	MTab	DAGOBAH	MantisTable	MTab	DAGOBAH	MantisTable	MTab	JenTab	DAGOBAH
		0.985	0.975	0.968	0.977	0.976	0.955	0.997	0.996	0.996
	R2-Bio	DAGOBAH	MTab	MantisTable	MTab	Magic	DAGOBAH	MTab	DAGOBAH	JenTab
		0.970	0.964	0.93	0.956	0.916	0.916	0.947	0.899	0.899
	R3-Biodiv	JenTab	MTab	DAGOBAH	KEPLER-aSI [4]	DAGOBAH	MTab	-	-	-
		0.602	0.522	0.496	0.593	0.391	0.123	-	-	-
	R3-Hard	DAGOBAH	MTab	MantisTable	DAGOBAH	MTab	MantisTable	MTab	JenTab	DAGOBAH
		0.974	0.968	0.959	0.99	0.984	0.965	0.993	0.992	0.991
	R3-Git (DBp)	-	-	-	DAGOBAH	KEPLER-aSI	MantisTable	-	-	-
		-	-	-	0.07	0.041	0.037	-	-	-
	R3-Git (Sch)	-	-	-	MantisTable	DAGOBAH	-	-	-	-
		-	-	-	0.205	0.183	-	-	-	-

a. Le score AH est pris en compte pour SemTab 2019 tandis que le score AF1 est utilisé pour SemTab 2020 et 2021

KGs publiques VS KGs ad-hocs. De nombreuses approches s'appuient sur des KG encyclopédiques tels que Wikidata et DBpedia, qui fournissent des informations riches permettant de produire des annotations de qualité. Toutefois, une plus grande quantité d'informations entraîne également une plus grande ambiguïté, et les bases de connaissances sont généralement incomplètes. Les évolutions des KGs dans le temps sont également un défi pour les approches. Nous observons que certaines approches [16, 19, 30, 49, 54, 40] traitent uniquement le KG cible comme un dictionnaire de concepts. Toutefois, savoir comment injecter correctement la structure du KG cible dans un modèle statistique reste une question ouverte.

6 Conclusion et directions de recherche

Ces dernières années ont été marquées par une croissance significative du domaine de l'interprétation de données tabulaires, notamment sous l'impulsion d'initiatives telles que SemTab. Dans cette étude, nous avons fourni un ensemble de définitions autour des données tabulaires et des tâches d'intepretation pour structurer et unifier le domaine ainsi qu'une vue actualisée sur les approches de l'état de l'art. Ces dernières sont classées en trois familles, les approches heuristiques, l'ingénierie de caractéristiques et l'apprentissage profond. Nous avons également mis en évidence les systèmes de STI les plus performants pour chaque ensemble de données et avons identifié plusieurs défis à relever pour améliorer les systèmes STI. Bien que les travaux récents aient permis de réaliser des progrès significatifs dans le domaine du STI, les approches existantes présentent plusieurs limites que nous décrivons ensuite.

Tout d'abord, la plupart des approches se concentrent sur des tables à sujet monocellulaire dans des tables relationnelles ou d'entités et font de fortes suppositions quant à la mise en forme utilisée pour la présentation des tables. En outre, les approches actuelles tiennent peu compte des spécificités de certaines tables relationnelles telles que les sujets cachés ou les sujets composés. Pour combler cette lacune et stimuler la recherche de nouvelles solutions, nous pensons qu'il est important d'élargir le spectre des complexités trouvé dans les corpus. À cette fin, nous recommandons de créer de nouveaux ensembles de données avec des structures de tableaux multiples et des contenus complexes afin d'aborder toute la diversité des données du monde réel. Nous estimons que la complexité du contenu ne devrait pas se limiter au bruit ajouté aux mentions, que ce soit de manière synthétique ou manuelle, car ce type de complexité peut être géré sans difficulté par la plupart des approches comme le démontre les résultats des derniers challenges SemTab.

Deuxièmement, les approches existantes supposent que le KG cible est complet et exempt d'erreurs. Par conséquent, une annotation (instance, type ou relation) peut toujours être générée même si le résultat correct ne se trouve pas dans le KG. Cette situation peut être préjudiciable, no-

tamment parce qu'elle peut propager l'erreur d'une annotation à l'ensemble de la colonne, voire à l'ensemble du tableau. Supposons par exemple un tableau avec une colonne contenant les noms de famille d'écrivains et une autre colonne contenant les titres de livres (pour les besoins de l'exemple, nous supposons que la majorité de ces livres ont été adaptés au cinéma). Si le KG cible couvre largement les films mais seulement quelques œuvres littéraires (ou est moins précis pour cette deuxième catégorie), le processus d'annotation pourrait typer la deuxième colonne comme "film", ce qui pourrait conduire à mal désambiguïser les mentions dans la première colonne (si certains acteurs apparentés ont des noms de famille similaires par exemple). En conséquence, ce tableau sera interprété comme un élément "acteurs-films" au lieu de la cible correcte "écrivains-livres". Certains mécanismes existants, tels que l'attribution d'un score de confiance à chaque candidat, peuvent aider à filtrer davantage les annotations incorrectes mais restent insuffisant. Enfin, nous soulignons que les approches futures devraient également envisager de s'attaquer à des domaines dans lesquels il n'existe que des KG naissants, l'objectif étant d'utiliser la STI pour augmenter ces KG dans une boucle vertueuse.

Troisièmement, nous observons que de nombreuses approches n'exploitent que partiellement les éléments du tableau (table 1), bien que les approches les plus récentes tendent à inverser cette tendance. Nous pensons que l'exploitation du plus grand nombre d'éléments possible devrait améliorer la précision en ajoutant davantage d'informations contextuelles. Ainsi, les modèles de langage pourraient être utilisés. En effet, on peut considérer un tableau comme un moyen de structurer le langage : dans le cas le plus simple, une ligne du tableau peut être considérée comme une phrase décrivant un sujet avec quelques attributs. Il en va de même pour le sous-graphe correspondant dans le KG cible. La représentation des phrases pourrait donc être utilisée pour calculer les similitudes. Néanmoins, la spécificité des données tabulaires et des KG doit être prise en compte, ce qui implique d'adapter les mécanismes d'attention à cette structure. Nous remarquons également que la plupart des approches traitent les tableaux de manière indépendante. Cependant, certaines tables sont liées les unes aux autres puisqu'elles peuvent être générées avec le même template, faire partie d'un corpus cohérent de tables ou être liées par des identifiants communs (e.g. bases de données SQL). Nous pensons que les systèmes de STI pourraient tirer un avantage significatif de la combinaison d'éléments de tableaux avec des connexions entre tableaux, qui peuvent être considérées comme une autre couche de contexte ajoutée pour capturer des informations plus riches sur les données à traiter.

Références

[1] Nora Abdelmageed and Sirko Schindler. JenTab Meets SemTab 2021's New Challenges. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[2] Ahmad Alobaid and Oscar Corcho. Balancing coverage and specificity for semantic labelling of subject columns. *Knowledge-Based Systems*, page 108092, 2022.

[3] Rabia Azzi and Gayo Diallo. AMALGAM : making tabular dataset explicit with knowledge graph. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, pages 9–16, 2020.

[4] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. KEPLER-asi at SemTab 2021. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[5] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. Tabel : Entity linking in web tables. In *14^{th} International Semantic Web Conference*, pages 425–441. Springer, 2015.

[6] Russa Biswas, Rima Türker, Farshad Bakhshandegan Moghaddam, Maria Koutraki, and Harald Sack. Wikipedia Infobox Type Prediction Using Embeddings. In *DL4KGS@ ESWC*, pages 46–55, 2018.

[7] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia-A crystallization point for the Web of Data. *Journal of web semantics*, 7(3) :154–165, 2009.

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase : a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data*, 2008.

[9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[10] Yoan Chabot, Thomas Labbe, Jixiong Liu, and Raphaël Troncy. DAGOBAH : an end-to-end context-free tabular data semantic annotation system. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching*, pages 41–48, 2019.

[11] Jiaoyan Chen, Ernesto Jimenez-Ruiz, Ian Horrocks, and Charles Sutton. ColNet : Embedding the Semantics of Web Tables for Column Type Prediction. In *33^{rd} AAAI International Conference on Artificial Intelligence*, 2018.

[12] Shuang Chen, Alperen Karaoglu, Carina Negreanu, Tingting Ma, Jin-Ge Yao, Jack Williams, Andy Gordon, and Chin-Yew Lin. Linkingpark : An integrated approach for semantic table interpretation. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[13] Marco Cremaschi, Roberto Avogadro, Andrea Barazzetti, and David Chieregato. MantisTable SE : an Efficient Approach for the Semantic Table Interpretation. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[14] Marco Cremaschi, Roberto Avogadro, and David Chieregato. MantisTable : an Automatic Approach for the Semantic Table Interpretation. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, pages 15–24, 2019.

[15] Dong Deng, Yu Jiang, Guoliang Li, Jian Li, and Cong Yu. Scalable column concept determination for web tables using large knowledge bases. In *PVLDB*, pages 1606–1617. VLDB Endowment, 2013.

[16] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. TURL : Table Understanding through Representation Learning. arXiv :2006.14806, 2020.

[17] Vasilis Efthymiou, Oktie Hassanzadeh, Mariano Rodriguez-Muro, and Vassilis Christophides. Matching web tables with knowledge base entities : from entity lookups to entity embeddings. In *16^{th} International Semantic Web Conference (ISWC)*, pages 260–277. Springer, 2017.

[18] Tong Guo, Derong Shen, Tiezheng Nie, and Yue Kou. Web table column type detection using deep learning and probability graph model. In *International Conference on Web Information Systems and Applications*, pages 401–414. Springer, 2020.

[19] Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. Sherlock : A deep learning approach to semantic data type detection. In *25^{th} ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pages 1500–1508, 2019.

[20] Viet-Phi Huynh, Yoan Chabot, Thomas Labbé, Jixiong Liu, and Raphaël Troncy. From Heuristics to Language Models : A Journey Through the Universe of Semantic Table Interpretation with DAGOBAH. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2022.

[21] Viet-Phi Huynh, Jixiong Liu, Yoan Chabot, Frédéric Deuzé, Thomas Labbé, Pierre Monnin, and Raphaël Troncy. DAGOBAH : Table and Graph Contexts for Efficient Semantic Annotation of Tabular Data. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[22] Viet-Phi Huynh, Jixiong Liu, Yoan Chabot, Thomas Labbé, Pierre Monnin, and Raphaël Troncy. DAGOBAH : Enhanced Scoring Algorithms for Scalable Annotations of Tabular Data. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[23] Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, and Kavitha Srinivas. SemTab 2019 : Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. In *European Semantic Web Conference (ESWC)*, pages 514–530. Springer, 2020.

[24] Emilia Kacprzak, José M Giménez-García, Alessandro Piscopo, Laura Koesten, Luis-Daniel Ibáñez, Jeni Tennison, and Elena Simperl. Making sense of numerical data-semantic labelling of web tables. In *European Knowledge Acquisition Workshop*, pages 163–178. Springer, 2018.

[25] Udayan Khurana and Sainyam Galhotra. Semantic annotation for tabular data, 2019.

[26] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. A large public corpus of web tables containing time and context metadata. In *25^{th} International Conference Companion on World Wide Web*, pages 75–76, 2016.

[27] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2) :1338–1347, 2010.

[28] Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. From tabular data to knowledge graphs : A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics*, page 100761, 2022.

[29] Jixiong Liu, Viet-Phi Huynh, Yoan Chabot, and Raphaël Troncy. Radar Station : Using KG Embeddings for Semantic Table Interpretation and Entity Disambiguation. In *21^{st} International Semantic Web Conference (ISWC)*, 2022.

[30] Pham Minh, Alse Suresh, A. Knoblock Craig, and Szekelyèle Pedro. Semantic Labeling : A Domain-Independent Approach. In *15^{th} International Semantic Web Conference (ISWC)*, pages 446—462, 2016.

[31] Hiroaki Morikawa. Semantic Table Interpretation using LOD4ALL. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, pages 49–56, 2019.

[32] Varish Mulwad, Tim Finin, and Anupam Joshi. Semantic message passing for generating linked data from tables. In *12^{th} International Semantic Web Conference (ISWC)*, pages 363–378. Springer, 2013.

[33] Varish Mulwad, Tim Finin, Zareen Syed, Anupam Joshi, et al. Using linked data to interpret tables. In *1^{st} International Workshop on Consuming Linked Data (COLD)*, 2010.

[34] Sebastian Neumaier, Jürgen Umbrich, Josiane Xavier Parreira, and Axel Polleres. Multi-level semantic labelling of numerical values. In *15^{th} International Semantic Web Conference (ISWC)*, pages 428–445, 2016.

[35] Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. MTab : Matching Tabular Data to Knowledge Graph using Probability Models. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2019.

[36] Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. TabEAno : table to knowledge graph entity annotation. arXiv :2010.01829, 2020.

[37] Phuc Nguyen, Ikuya Yamada, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. Mtab4wikidata at semtab 2020 : Tabular data annotation with wikidata. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[38] Phuc Nguyen, Ikuya Yamada, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. SemTab 2021 : Tabular Data Annotation with MTab Tool. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[39] Daniela Oliveira and Mathieu d'Aquin. Adog-annotating data with ontologies and graphs. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2019.

[40] S Krishnamurthy Ramnandan, Amol Mittal, Craig A Knoblock, and Pedro Szekely. Assigning semantic labels to data sources. In *European Semantic Web Conference (ESWC)*, pages 403–417. Springer, 2015.

[41] Dominique Ritze and C. Bizer. Matching Web Tables To DBpedia - A Feature Utility Study. In *International Conference on Extending Database Technology (EDBT)*, pages 210—-221, 2017.

[42] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching html tables to dbpedia. In 5^{th} *International Conference on Web Intelligence, Mining and Semantics*, pages 1–6, 2015.

[43] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching HTML Tables to DBpedia. In 5^{th} *International Conference on Web Intelligence, Mining and Semantics (WIMS)*, pages 1–6, 2015.

[44] Yoones A Sekhavat, Francesco Di Paolo, Denilson Barbosa, and Paolo Merialdo. Knowledge base augmentation using tabular data. In *LDOW*, 2014.

[45] Renat Shigapov, Philipp Zumstein, Jan Kamlah, Lars Oberländer, Jörg Mechnich, and Irene Schumm. bbw : Matching CSV to Wikidata via Meta-lookup. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, volume 2775, pages 17–26. RWTH, 2020.

[46] Gaurav Singh, Siffi Singh, Joshua Wong, and Amir Saffari. Relation Extraction from Tables using Artificially Generated Metadata. arXiv :2108.10750, 2021.

[47] Bram Steenwinckel, Filip De Turck, and Femke Ongeane. MAGIC : Mining an Augmented Graph using INK, starting from a CSV. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[48] Bram Steenwinckel, Gilles Vandewiele, Filip De Turck, and Femke Ongenae. Csv2kg : Transforming tabular data into semantic knowledge. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2019.

[49] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. Annotating Columns with Pre-trained Language Models. arXiv :2104.01785, 2021.

[50] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. YAGO 4 : A Reason-able Knowledge Base. In *European Semantic Web Conference (ESWC)*, pages 583–596. Springer, 2020.

[51] Avijit Thawani, Minda Hu, Erdong Hu, Husain Zafar, Naren Teja Divvala, Amandeep Singh, Ehsan Qasemi, Pedro A Szekely, and Jay Pujara. Entity Linking to Knowledge Graphs to Infer Column Types and Properties. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, volume 2019, pages 25–32, 2019.

[52] Petros Venetis, Alon Y Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, and Gengxin Miao. Recovering semantics of tables on the web. *PVLDB*, 4(9) :528–538, 2011.

[53] Denny Vrandečić and Markus Krötzsch. Wikidata : a free collaborative knowledge base. *Communications of the ACM*, 57(10) :78–85, 2014.

[54] Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. Tcn : Table convolutional network for web table interpretation. arXiv :2102.09460, 2021.

[55] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q Zhu. Understanding tables on the web. In *International Conference on Conceptual Modeling*, pages 141–155. Springer, 2012.

[56] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI Conference on Artificial Intelligence*, 2014.

[57] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. Sato : Contextual Semantic Type Detection in Tables, 2019.

[58] Shuo Zhang, Edgar Meij, Krisztian Balog, and Ridho Reinanda. Novel entity discovery from web tables. In *The Web Conference*, pages 1298–1308, 2020.

[59] Ziqi Zhang. Effective and efficient semantic table interpretation using tableminer+. *Semantic Web*, 8(6) :921–957, 2017.

[60] Yiwei Zhou, Siffi Singh, and Christos Christodoulopoulos. Tabular Data Concept Type Detection Using Star-Transformers. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3677–3681, 2021.

[61] Stefan Zwicklbauer, Christoph Einsiedler, Michael Granitzer, and Christin Seifert. Towards Disambiguating Web Tables. In *International Semantic Web Conference (Posters & Demos)*, pages 205–208, 2013.

Et si on comprenait la structure de graphes de connaissances comme Wikidata ?

Hassan Abdallah[1], Béatrice Markhoff[2], Arnaud Soulet[1]

[1] Université de Tours, EA 6300 LIFAT, Blois
[2] Université de Tours, UMR 7324 CITERES, Blois

prenom.nom@univ-tours.fr

Résumé

La production participative de graphes de connaissances comme Wikidata a permis l'émergence de vastes bases de connaissances agglomérant des expertises et des opinions variées. Fortement dépendant de sa communauté, ce processus décentralisé interroge sur sa capacité à faire émerger un graphe de connaissances représentatif et cohérent. Dans cet article, nous affirmons que la représentation des connaissances gagnerait à modéliser la manière dont les faits s'organisent dans la partie assertionnelle. L'objectif serait de définir et étudier des processus permettant de générer des données synthétiques qui ressemblent étroitement aux graphes de connaissances réels. Nous indiquons des retombées possibles de ces modèles en optimisation et en analyse de données. Enfin, nous envisageons les principaux verrous scientifiques à lever pour parvenir à modéliser la structure des graphes de connaissances.

Mots-clés

Graphe de connaissances, modèle génératif, Wikidata.

Abstract

Crowdsourcing of knowledge graphs (KGs) such as Wikidata has allowed the emergence of vast knowledge bases agglomerating various expertises and opinions. Strongly dependent on its community, this decentralized process questions its capacity to produce a representative and coherent KG. In this paper, we argue that knowledge representation would benefit from modeling the way facts are organized in the assertional part. The goal would be to define and study processes to generate synthetic graphs that closely resemble real KGs. We indicate possible implications of these models in optimization and KG analysis. Finally, we consider the main scientific obstacles to overcome in order to model KGs.

Keywords

Knowledge graph, generative model, Wikidata.

1 Introduction

La modélisation de la structure des réseaux et de leur dynamique est un domaine qui a pris de l'ampleur dans les années 2000. Elle s'intéresse aux relations et interconnexions entre des objets, dans des domaines aussi divers que les réseaux d'ordinateurs ou la biologie. En particulier, elle a apporté un ensemble important de connaissances sur l'émergence de structures et leurs dynamiques dans les artefacts du Web : réseau des pages HTML et réseaux sociaux [6], résultats de systèmes distribués d'annotations collaboratives (folksonomies) [19]. Parmi les caractéristiques communes à ces artefacts, il y a le fait d'être construits par des ensembles de personnes qui interagissent de façon distribuée, ne communiquant qu'à travers leurs actions sur des ressources du Web et, selon les cas, se coordonnant en communautés ou pas. L'étude de la structure des artefacts résultant de ces actions isolées, qui deviennent des interactions par le fait d'agir sur les mêmes ressources du Web (pointer vers une page Web depuis une autre, suivre ou répondre à un compte de réseau social, annoter la même ressource) apporte divers éclairages et permet des analyses riches (voir par exemple [23] sur la visualisation des structures et dynamiques de connaissances contenues dans des articles scientifiques). Notre intuition est que des modèles génératifs des graphes de connaissances du Web offriraient des perspectives de réflexion tout aussi riches, et auraient par ailleurs des applications concrètes pour améliorer l'exploitation de ces derniers. Ceci nous paraît particulièrement vrai pour les grands graphes construits de manière participative, de façon distribuée et peu contrainte, comme Wikidata. Un outil qui reproduirait une telle construction d'une connaissance commune, en d'autres termes une construction de consensus sur des connaissances, permettrait évidemment d'observer et analyser « in vitro » cette construction, de produire des benchmarks plus représentatifs, et d'élaborer des méthodes d'analyse statistique et d'apprentissage plus fiables.

L'étude de la structure des graphes de connaissances et de son évolution est donc impérative pour gagner en compréhension sur les bases de connaissances produites collaborativement. De manière immédiate, il est possible d'appliquer à ces graphes de connaissances des statistiques descriptives pour observer des phénomènes [13]. L'une des principales caractéristiques d'un graphe de connaissances est sa distribution de degrés, qui mesure le nombre de faits impliquant une entité avec les autres entités du graphe. De manière intéressante, cette information précieuse indique la répartition des faits au sein du graphe, mettant en lu-

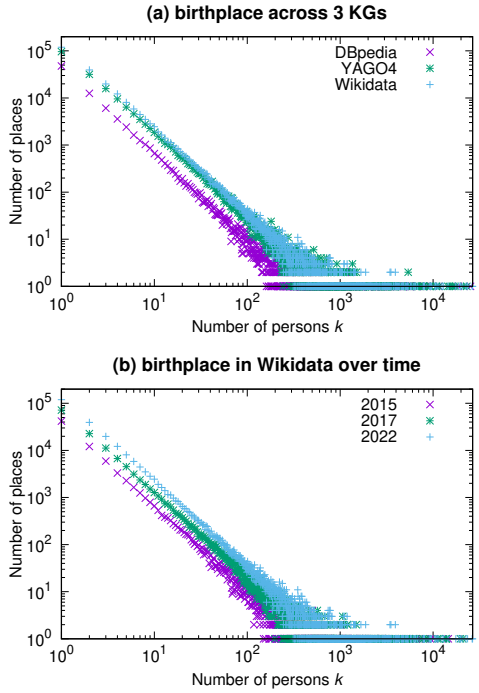


FIGURE 1 – Illustration de l'existence d'une structure pour la relation `place of birth` (dénotée par `wdt:P19`) (a) à travers 3 graphes de connaissances issus de productions collaboratives et (b) au fil du temps dans Wikidata.

mière des entités mal-renseignées et celles concentrant la majorité des connaissances. Prenons par exemple la relation `place of birth` (dénotée par `wdt:P19` dans Wikidata) présente dans DBpedia [5], YAGO4 [26] et Wikidata [40], la figure 1(a) représente le nombre de lieux en fonction du nombre de naissances pour ces 3 graphes. On y voit que dans ces trois ressources il y a de l'ordre de 10^5 lieux dans lesquels seulement une personne est déclarée être née (en haut à gauche). On voit également qu'il y a une concentration des déclarations de naissance sur relativement peu de lieux (en bas à droite, entre 100000 et plusieurs dizaines de millions pour une dizaine de villes). Étonnamment, bien que ces trois graphes ne portent pas sur les mêmes données, ils exhibent une structure similaire pour cette relation. De plus, la figure 1(b) représente la même distribution pour Wikidata en considérant 3 années différentes depuis sa création. Outre l'augmentation attendue du nombre de faits, nous constatons que la structure à travers le temps reste la même. Nous avons également observé cette même stabilité de la structure à travers le temps et les graphes pour d'autres relations : `instance of` (`wdt:P31`), `subclass of` (`wdt:P279`), `creator` (`wdt:P170`), etc. A la lumière de ces exemples, il parait nécessaire d'aller au-delà de simples observations statistiques pour comprendre comment s'accumulent les faits au sein de ces graphes de connaissances et les phénomènes sous-jacents conduisant à l'émergence de structures régulières.

Positionnement La représentation des connaissances [38] s'intéresse en profondeur à la formalisation et la modélisation de la partie terminologique avec notamment des travaux variés sur les logiques de description et les ontologies. Cette compréhension de la partie terminologique se répercute sur la forme des faits constituant la partie assertionnelle, mais elle ne décrit pas certaines singularités de la structuration des entités et des relations (comme la distribution des faits). De plus, il serait impossible d'étudier Wikidata à partir de son ontologie puisque ce graphe de connaissances ne repose pas sur une ontologie [41]. Nous pensons donc qu'il est nécessaire d'étendre le champs de la représentation des connaissances à l'étude de la manière dont s'organisent les faits pour constituer le graphe de connaissances. Il s'agit de développer des modèles pour capturer l'organisation des faits afin de pouvoir générer des données synthétiques qui ressemblent étroitement aux graphes de connaissances réels, à l'instar des travaux qui ont été menés pour expliquer la structuration du Web [6] ou celle des folksonomies [19]. En effet, l'objectif d'un modèle génératif est de reproduire les données réelles ; donc, si cette reproduction est fidèle (si le modèle est correct), alors l'étude de la structure générée par le modèle permet de découvrir des caractéristiques de la structure du graphe réel, et de son évolution. Pour étayer ce positionnement, nous rappelons en section 2 ce qui caractérise Wikidata d'une part, et d'autre part les travaux existants sur la compréhension des graphes de connaissances, puis nous nous intéressons aux retombées qu'aurait une modélisation de la structure des graphes de connaissances dans la section 3. Ensuite, nous envisageons les principaux verrous scientifiques à lever pour parvenir à modéliser les graphes de connaissances, dans la section 4, avec plusieurs directions de recherche.

2 Genèse de Wikidata et compréhension des graphes de connaissances

L'émergence des grands graphes de connaissances du Web, et en particulier de Wikidata, repose en grande partie sur la production collaborative, ce qui soulève la question de la légitimité des données construites en termes de représentativité et de cohérence. En effet, la production collaborative est un processus qui consiste à solliciter les contributions d'un grand groupe de personnes pour obtenir des données ou des informations [36]. Ce processus a gagné en popularité avec la croissance des technologies numériques facilitant la mise en relation de personnes du monde entier pour qu'elles puissent travailler ensemble à la réalisation d'un objectif commun. La production collaborative est particulièrement utile lorsque de grandes quantités de données ou un large éventail de perspectives sont nécessaires [32]. D'une part, les données ainsi produites peuvent être vastes et variées (e.g., textes ou images) avec la possibilité de rassembler de grands volumes en un temps relativement court. D'autre part, un autre avantage de la production collaborative est la potentielle diversité des perspectives exprimées. En sollicitant les contributions d'un grand nombre

de personnes, cela permet de s'assurer qu'un large éventail d'opinions, d'expériences et de points de vue sont représentés dans les données produites. Dans le cas de Wikidata [40], cet aspect s'avère particulièrement important. En s'appuyant sur les contributions d'une communauté diversifiée de contributeurs, Wikidata est en mesure d'élargir et d'affiner sa base de connaissances au fil du temps pour renforcer l'exactitude et l'exhaustivité des données. En revanche, la décentralisation de la génération des faits et leur maintenance par une large communauté d'individus soulève des incertitudes. Malgré la diversité désirée de ses contributeurs, la communauté peut être déséquilibrée introduisant des biais sur les données produites. Le volume important de données rassemblées peut négliger certains domaines, entraînant des biais de représentativité culturels ou sociaux [12]. De plus, pour un même domaine polémique, il est possible d'imaginer que sans organisation centralisée les divergences entre les points de vue des individus provoquent des inconsistances voire de l'instabilité. Par exemple, sans ontologie de référence, on peut s'interroger sur la convergence de la terminologie de Wikidata, co-construite par les contributeurs. De nombreux travaux se sont intéressés aux aspects sociaux et organisationnels de la communauté collaborant pour construire un tel graphe de connaissances [34, 28]. Mais, à notre connaissance, la répercussion de cette production collaborative sur l'émergence d'une structuration des données n'a pas reçu d'attention.

La plupart des approches de construction de graphes de connaissances consistent en des workflows de production partant de diverses sources de données (textes, pages web, tableurs, bases de données sous toutes les formes) et utilisant une ou plusieurs ontologies, ainsi qu'un ou plusieurs thésauri de référence. Les initiateurs et pilotes de Wikidata ont délibérément choisi de ne pas construire une ontologie au préalable et de ne pas imposer de thesaurus non plus, sachant que Wikipedia en a fourni le noyau de départ [41]. Pour autant, cela reste aux ontologies, ou schémas, que l'on pense quand il est question de la structure des graphes de connaissances [20]. Au-delà de la nécessité pour les utilisateurs de découvrir et comprendre les ontologies utilisées dans les graphes de connaissances, comme il est toujours difficile de saisir rapidement le contenu de ces graphes, de nombreux travaux s'attachent à en extraire des informations statistiques [7], des résumés [10, 17], des schémas [21], des profils [35, 14], des contraintes de forme [30], ou bien à les munir d'interfaces d'interrogation aussi intuitives que possible, par exemple [24, 39] pour Wikidata. En parallèle à ces efforts pour exhiber la sémantique des données contenues dans les graphes de connaissances, il y a également de très nombreuses propositions consistant à aborder les graphes de connaissances avec des outils de deep learning [25, 31]. Globalement ces modèles génératifs profonds permettent une reproduction fine mais nécessitent des données d'apprentissage et apportent peu de compréhension des graphes car ils requièrent trop de paramètres. Malgré la diversité des approches que nous venons de résumer, nous n'avons pas trouvé de proposition visant à comprendre et modéliser la topologie propre aux graphes de connaissances.

3 Intérêt d'une modélisation

Il ne fait aucun doute que la proposition de modèles précis pour les graphes de connaissances apporterait énormément au domaine tant d'un point de vue pratique que théorique. Au-delà de la création de données synthétiques, l'analyse théorique du modèle génératif peut en effet apporter une loi de probabilité sur les données. Par exemple, la distribution du nombre de naissances de la figure 1 suit une loi de puissance d'exposant 1.91 dans Wikidata. La connaissance d'une telle loi est évidemment utile à diverses fins notamment en ingénierie des données, en analyse de données et en qualification des données.

3.1 Optimisation et benchmarking

Comme la taille des graphes de connaissances les plus importants comme Wikidata ne cesse de croître, il est important d'améliorer la performance des systèmes d'interrogation comme les moteurs SPARQL [1, 3]. A l'instar des bases de données, il est nécessaire d'exploiter les propriétés des données pour optimiser l'exécution des requêtes et de s'appuyer sur des benchmarks pour comparer les systèmes. D'une part, les propriétés mathématiques dérivées des modèles peuvent être injectées directement dans le système d'interrogation à la place des statistiques sur les données. En effet, le stockage de données et l'optimisation du plan d'exécution nécessitent des informations sur la répartition des données pour améliorer l'exécution des requêtes. Ces statistiques sont coûteuses à obtenir et à maintenir voire sont remplacées par des heuristiques imprécises [37]. A l'inverse, les modèles reposent sur des paramètres d'entrée qui résument précisément les principales caractéristiques des données réelles à simuler (e.g., l'exposant 1.91 indique la répartition globale des faits de la relation `place of birth`). D'autre part, le développement et le test des systèmes d'interrogation nécessitent aussi des benchmarks variés pour analyser les différents contextes d'interrogation. La génération de données synthétiques est une approche qui permet de relever ce défi. Elle consiste à créer de nouvelles données ayant des caractéristiques statistiques similaires aux données réelles, ce qui permet aux chercheurs de tester et d'optimiser leurs propositions sans avoir recours à des données réelles. Cette approche est particulièrement utile pour pouvoir observer la répercussion d'un paramètre précis de la génération de données sur le système d'interrogation pour mieux comprendre ses forces et ses faiblesses. A notre connaissance, les principaux générateurs de données synthétiques pour les graphes de connaissances utilisent uniquement des lois normale ou uniforme pour générer la distribution des faits comme BSBM [8] ou LUBM [18]. Il est clair que ces benchmarks ne sont pas adaptés pour simuler des relations comme `place of birth` se rapprochant davantage d'une loi de puissance. Par conséquent, la proposition de modèles fins pour générer des données synthétiques est cruciale pour construire des benchmarks plus réalistes.

3.2 Exploration de données

Nous pensons que la connaissance de modèles pourrait bénéficier à la fois aux méthodes d'exploration de données descriptive et prédictive. De manière évidente, les statistiques descriptives s'appuient sur les lois statistiques connues pour les observations. A la lumière de la figure 1, il semble judicieux d'utiliser une loi de pareto plutôt qu'une loi normale pour analyser le nombre de naissances par ville. De la même manière, si l'on souhaite classifier les villes en différents groupes, la distribution suggère d'éviter d'utiliser K-means [22], plutôt adapté à un mélange de gaussiennes. La difficulté est que la distribution dépend de chaque relation, renforçant l'intérêt de disposer d'un modèle qui permettrait de les distinguer. En outre, l'objectif d'un modèle génératif pour les graphes de connaissances est aussi de simuler la croissance du graphe dans le temps. Par analogie avec d'autres travaux en science des réseaux, il serait alors possible d'estimer la probabilité pour une entité de recevoir un nouveau fait en se basant uniquement sur la structure du graphe. Cette connaissance s'avère évidemment précieuse pour analyser des données. La prédiction de lien est une tâche prédictive très populaire pour compléter et augmenter les graphes de connaissances [31]. Les approches actuelles basées sur des indices locaux pourraient alors prendre en compte la structure globale du graphe. Inversement, de nombreuses méthodes en détection d'anomalie [2] exploitent une connaissance structurelle attendue sur les graphes pour identifier comme anomalies les arcs ou les noeuds qui dévient de cette structure. Schématiquement, pour la relation `place of birth`, il est possible de modéliser le nombre de naissances nouvelles rattachées annuellement à chaque ville. Dès lors, une ville qui recevrait subitement de nombreux faits au quotidien contre une estimation attendue de quelques faits sur l'année, serait identifiée comme une anomalie.

3.3 Qualification des données

Une question critique est de déterminer si la structure du graphe de connaissances est stable. Il est essentiel de comprendre si la distribution de probabilité des degrés dans le graphe de connaissances converge ou non [19]. En analysant cette caractéristique du graphe, il est possible d'avoir une idée de la stabilité générale du graphe et de la probabilité qu'il évolue au fil du temps [16]. La représentativité de la connaissance est une autre caractéristique critique qui doit être prise en considération lors de l'exploitation de graphes de connaissances. Il s'agit de savoir dans quelle mesure le graphe reliant des entités est complet et non biaisé. Un modèle adéquat de la structure du graphe et de son évolution permettrait de s'assurer de sa représentativité en identifiant le nombre minimum d'entités requis pour la garantir [33]. Par ailleurs, un modèle génératif peut aider à déterminer les entités vulnérables ou au contraire robustes dans le graphe de connaissances, en déterminant si l'ajout ou la suppression d'un fait peut remettre en cause la connaissance courante. Un autre aspect encore, qui peut être analysé à l'aide d'un modèle de la structure du graphe, est la découverte de nouvelles relations entre des entités, qui peut émerger d'interactions entre les contributeurs, ce qui apporterait un éclairage complémentaire sur la dynamique sociale de l'ingénierie collaborative des connaissances [28, 27]. De plus, un modèle peut permettre de détecter des erreurs ou incohérences dans les données. Par exemple, dans la figure1(a), il est clair que les trois sources sont concordantes, aussi un modèle de cette relation pourrait montrer des contradictions dans les informations provenant d'une autre source, concernant cette même relation. Cela aiderait à améliorer la qualité des données du graphe [29] et le rendrait ainsi plus utile à différentes applications. Par conséquent, un modèle bien conçu permettrait de montrer des caractéristiques non apparentes et des relations implicites, menant à de nouvelles idées et découvertes.

4 Défis de la modélisation

Pour modéliser la structuration de la production collaborative de connaissances, il serait possible d'utiliser des techniques d'analyse de réseaux comme initié par [6]. Ces techniques fournissent un moyen de décrire les relations entre les entités, et de mettre en évidence des motifs d'interactions et de collaboration qui apparaissent au sein de communautés de contributeurs et contributrices. De nombreux travaux ont suivi ceux de Albert et Barabási [6], encore tout récemment un nouveau modèle générique a été proposé dans [16] pour reproduire la distribution de différents réseaux complexes [11] en focalisant sur la fonction d'attachement pour la connexion des noeuds. Cette fonction gouverne la manière dont les nouveaux noeuds sont connectés au graphe existant. Un autre modèle de croissance a été introduit dans [9] pour décrire les graphes *dirigés* sans-échelle dont la taille augmente grâce à une fonction d'attachement préférentiel.

Nous n'avons trouvé pour les graphes de connaissances aucune proposition de modélisation de la structure et de la dynamique du graphe en lui-même. Ceci s'explique sans doute par un ensemble de caractéristiques propres aux graphes de connaissance, qui font que les modèles existants ne peuvent pas s'y appliquer, ni même s'y adapter simplement. La principale tient sans doute au fait qu'ils représentent des connaissances, et qui plus est, pour un graphe comme celui de Wikidata, des connaissances dans des domaines très divers. Ces connaissances sont donc décrites par *des relations très diverses*, dont le nombre augmente avec la croissance du graphe. Même si leur création est nettement plus encadrée que la création d'entités, il n'en demeure pas moins qu'en 2015 il y avait de l'ordre de 500 relations directes (nous ne considérons pas celles qui permettent de caractériser les assertions) et en 2022 il y en a pratiquement trois fois plus. Alors que les modèles existant pour les réseaux considèrent tous les arcs du graphe de façon indistincte, pour un graphe de connaissances il est crucial que le modèle représente précisément le comportement des ensembles d'arcs qui ont la même étiquette, correspondant à une relation décrivant un ensemble d'entités du monde réel.

Par conséquent concevoir un modèle génératif fiable pour des graphes de connaissances est un vrai défi. Pour s'y atta-

quer il faut prendre en compte plusieurs dimensions, parmi lesquelles la véracité, le volume et la variété des données de ces graphes.

Véracité : Concernant la véracité des données, la conception d'un modèle robuste d'un graphe de connaissances doit prendre en compte le fait que les connaissances présentes sont fréquemment incomplètes ou, plus rarement, erronées [33], ce qui rend nécessaire de développer un modèle capable de traiter des éléments manquants ou bruités. Le modèle doit aussi pouvoir rendre compte de la nature dynamique du monde représenté, par exemple l'objet de la relation `wdt:P39` (`position held`) change régulièrement pour une personne donnée. Ces évolutions du graphe, différentes de simples ajouts d'entités, doivent être modélisées aussi. De plus, tandis que la plupart des modèles existants s'attachent à ajouter des entités, pour les graphes de connaissances il faudrait également modéliser l'évolution inverse, la suppression d'entités (par exemple des entités devenues obsolètes).

Volume de données : L'une des principales difficultés réside dans le très grand volume de données des graphes comme Wikidata. Or pour mener les expérimentations nécessaires au paramétrage d'un modèle génératif, il est nécessaire de recueillir les données du graphe par des requêtes analytiques. Cela nécessite d'utiliser des algorithmes capables de traiter ces très grands volumes de données, sans compromettre leur performance. De plus, le modèle en lui-même doit être efficace dans la génération de gros graphes synthétiques, ce qui souleve des défis algorithmiques [4]. Pour la mise au point du modèle comme pour son exécution, la gestion de la mémoire est un autre problème crucial à considérer.

Variété : La modélisation de graphes de connaissances qui contiennent des connaissances multidisciplinaires, à l'image de Wikidata, est une tâche compliquée qui nécessiterait une approche interdisciplinaire. En d'autres termes, lors de la génération des données synthétiques, il serait nécessaire de vérifier la capacité du modèle à fournir une représentation pertinente de ces connaissances du monde réel. Dans les différents domaines représentés par les graphes de connaissances, certains bénéficient déjà de modèles bien connus, par exemple en bibliométrie [15], tandis que d'autres pas. Mettre au point un modèle pour un domaine requiert une connaissance de ce domaine et des méthodes statistiques pour générer des valeurs fiables. Cependant, créer un modèle unique et spécifique à chaque domaine n'est pas faisable. Aussi, il faudrait un modèle capable de fonctionner pour les diverses disciplines et domaines.

De plus, notons que les graphes de connaissances sont constitués de différents types d'entités et de relations, en particulier des entités correspondant à l'observation du réel et des entités plus conceptuelles, décrivant des connaissances dérivées des données d'observation. Prenons l'exemple d'un graphe de connaissances en biologie médicale. Ce graphe peut inclure à la fois des entités réelles, telles que les maladies et les symptômes, et des entités conceptuelles, telles que les mécanismes biologiques sous-jacents à l'origine de ces maladies. La diversité des entités et des relations, ainsi que la variété des domaines et disciplines, rendent difficile la création d'un modèle de croissance aléatoire qui refléterait avec précision la mosaïque structurelle des graphes de connaissances.

5 Conclusion

Cet article propose de s'intéresser à la structure des graphes de connaissances en concevant et exploitant des modèles génératifs inspirés de la science des réseaux. Leur mise au point soulève de nombreux défis liés à la forme des graphes de connaissances bien plus complexe, avec la superposition des sémantiques portées par chacune des relations. Pourtant, de tels modèles singeant les données réelles seraient idéals pour mieux comprendre la structure sous-jacente des graphes et leur dynamique. Cette compréhension serait utile pour de nombreux travaux sur les graphes de connaissances allant de l'optimisation à la qualification des données. Au-delà, comme le graphe est un miroir des connaissances d'un domaine, elle permettrait aussi d'apporter un éclairage sur la constitution des connaissances au sein de ce domaine.

Références

[1] Ibrahim Abdelaziz, Razen Harbi, Zuhair Khayyat, and Panos Kalnis. A survey and experimental comparison of distributed sparql engines for very large rdf data. *VLDB*, 10(13) :2049–2060, 2017.

[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description : a survey. *Data mining and knowledge discovery*, 29 :626–688, 2015.

[3] Waqas Ali, Muhammad Saleem, Bin Yao, Aidan Hogan, and Axel-Cyrille Ngonga Ngomo. A survey of rdf stores & sparql engines for querying knowledge graphs. *The VLDB Journal*, pages 1–26, 2022.

[4] James Atwood, Bruno Ribeiro, and Don Towsley. Efficient network generation under general preferential attachment. In *WWW*, pages 695–700, 2014.

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia : A nucleus for a web of open data. In *ISWC*, pages 722–735, 2007.

[6] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439) :509–512, 1999.

[7] Mohamed Ben Ellefi, Zohra Bellahsene, John G Breslin, Elena Demidova, Stefan Dietze, Julian Szymański, and Konstantin Todorov. Rdf dataset profiling–a survey of features, methods, vocabularies and applications. *Semantic Web*, 9(5) :677–705, 2018.

[8] Christian Bizer and Andreas Schultz. The berlin sparql benchmark. *International Journal on Semantic Web and Information Systems*, 5(2) :1–24, 2009.

[9] Béla Bollobás, Christian Borgs, Jennifer T Chayes, and Oliver Riordan. Directed scale-free graphs. In *SODA*, volume 3, pages 132–139, 2003.

[10] S. Cebiric, F. Goasdoue, H. Kondylakis, D. Kotzinos, I. Manolescu, G. Troullinou, and M. Zneika. Summarizing Semantic Graphs : A survey. *The VLDB Journal*, 28 :295–327, 2018.

[11] Fan Chung, Fan RK Chung, Fan Chung Graham, Linyuan Lu, et al. *Complex graphs and networks*. Number 107. American Mathematical Soc., 2006.

[12] Gianluca Demartini. Implicit bias in crowdsourced knowledge graphs. In *WWW*, pages 624–630, 2019.

[13] Li Ding and Tim Finin. Characterizing the semantic web on the web. In *ISWC*, pages 242–257, 2006.

[14] Lamine Diop, Béatrice Markhoff, and Arnaud Soulet. TTProfiler : types and terms profile building for online cultural heritage knowledge graphs. *JOCCH*, 2023.

[15] Leo Egghe. *Power laws in the information production process : Lotkaian informetrics*. 2005.

[16] Frédéric Giroire, Stéphane Pérennes, and Thibaud Trolliet. A random growth model with any real or theoretical degree distribution. *Theoretical Computer Science*, 940 :36–51, 2023.

[17] F. Goasdoue, P. Guzewicz, and I. Manolescu. RDF graph summarization for first-sight structure discovery. *The VLDB Journal*, 29(5) :1191–1218, 2020.

[18] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. Lubm : A benchmark for owl knowledge base systems. *Journal of Web Semantics*, 3(2-3) :158–182, 2005.

[19] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *WWW*, pages 211–220, 2007.

[20] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4) :1–37, 2021.

[21] Kenza Kellou-Menouer, Nikolaos Kardoulakis, Georgia Troullinou, Zoubida Kedad, Dimitris Plexousakis, and Haridimos Kondylakis. A survey on semantic schema discovery. *The VLDB Journal*, pages 1–36, 2021.

[22] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2) :129–137, 1982.

[23] Quentin Lobbé, Alexandre Delanoë, and David Chavalarias. Exploring, browsing and interacting with multi-level and multi-scale dynamics of knowledge. *Information Visualization*, 21(1) :17–37, 2022.

[24] Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. Getting the most out of Wikidata : semantic technology usage in wikipedia's knowledge graph. In *ISWC*, pages 376–394, 2018.

[25] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proc. of the IEEE*, 104(1) :11–33, 2015.

[26] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. Yago 4 : A reason-able knowledge base. In *ESWC*, pages 583–596, 2020.

[27] Guangyuan Piao and Weipeng Huang. Learning to predict the departure dynamics of Wikidata editors. In *ISWC*, pages 39–55, 2021.

[28] Alessandro Piscopo and Elena Simperl. Who models the world ? collaborative ontology creation and user roles in Wikidata. *HCI*, 2 :1–18, 2018.

[29] Alessandro Piscopo and Elena Simperl. What we talk about when we talk about Wikidata quality : a literature survey. In *the 15th International Symposium on Open Collaboration*, pages 1–11, 2019.

[30] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Extraction of validating shapes from very large knowledge graphs. *PVLDB*, 16(5) :1023–1032, 2023.

[31] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction : A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, 15(2) :1–49, 2021.

[32] Cristina Sarasua, Elena Simperl, Natasha F Noy, Abraham Bernstein, and Jan Marco Leimeister. Crowdsourcing and the semantic web : A research manifesto. *Human Computation*, 2(1), 2015.

[33] Suhas Shrinivasan and Simon Razniewski. How stable is knowledge base knowledge ? *arXiv preprint arXiv :2211.00989*, 2022.

[34] Elena Simperl and Markus Luczak-Rösch. Collaborative ontology engineering : a survey. *The Knowledge Engineering Review*, 29(1) :101–131, 2014.

[35] Blerina Spahiu, Riccardo Porrini, Matteo Palmonari, Anisa Rula, and Andrea Maurino. ABSTAT : Ontology-Driven Linked Data Summaries with Pattern Minimalization. In *ESWC*, pages 381–395, 2016.

[36] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[37] Petros Tsialiamanis, Lefteris Sidirourgos, Irini Fundulaki, Vassilis Christophides, and Peter Boncz. Heuristics-based query optimisation for sparql. In *ICDT/EBDT*, pages 324–335, 2012.

[38] Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. *Handbook of knowledge representation*. Elsevier, 2008.

[39] Hernán Vargas, Carlos Buil-Aranda, Aidan Hogan, and Claudia López. A user interface for exploring and querying knowledge graphs (extended abstract). In *IJCAI*, pages 4785–4789, 2020.

[40] Denny Vrandečić and Markus Krötzsch. Wikidata : a free collaborative knowledgebase. *Communications of the ACM*, 57(10) :78–85, 2014.

[41] Denny Vrandečić, Lydia Pintscher, and Markus Krötzsch. Wikidata : The making of. In *the ACM Web Conference 2023*, pages 615–624, 2023.

Session 6 : Ingénierie de connaissances, Données FAIR

Connexions et relations

G. Kassel

Laboratoire MIS, Université de Picardie Jules Verne
33 rue Saint-Leu, 80039 Amiens Cedex 1

Gilles.kassel@u-picardie.fr

Résumé

Dans cet article, nous poursuivons la définition d'une espèce d'ontologies baptisées « ontologies épistémiques » en caractérisant la nature des entités complexes *qu'elles représentent. Parmi ces entités complexes figurent les propositions et événements (dans la sphère mentale) et les états de choses (dans la sphère physique). Pour rendre compte de leur structure, nous faisons appel à la figure ontologique de la '*connexion'*. Dans l'article, nous privilégions l'étude des états de choses, que nous identifions à des connexions internes (liant les objets et processus à leurs qualités) et à des connexions externes (entre objets et processus). De telles connexions sont récurrentes et nous avons connaissance de ces répétitions sous la forme de types, ou liens généraux, de connexion. Nous définissons ces liens généraux de connexion comme une espèce de relations.*

Mots-clés

Ontologie fondatrice, ontologie épistémique, bipartition des entités physiques et mentales, connexions, relations

Abstract

In this paper, we further define a species of ontologies called "epistemic ontologies" by characterizing the nature of the complex entities they represent. Among these complex entities are propositions and events (in the mental sphere) and states of affairs (in the physical sphere). To account for their structure, we use the ontological figure of 'connection'. In the article, we privilege the study of states of affairs, which we identify with internal connections (linking objects and processes to their qualities) and external connections (between objects and processes). Such connections are recurrent and we have knowledge of these repetitions in the form of types, or general links, of connection. We define these general links as a kind of relationship.

Keywords

Foundational ontology, epistemic ontology, bipartition of physical and mental entities, connections, relations

1 Introduction

Cet article vise à approfondir le cadre métaphysique que nous avons proposé récemment pour déployer des ontologies qualifiées d'*épistémiques* et définies comme des systèmes de catégories représentant des *objets de représentation* ou *objets de connaissance* du monde [16] (*cf.* Fig. 1 pour une esquisse d'une ontologie fondatrice épistémique). L'étude que nous menons porte tout particulièrement sur la caractérisation des entités *complexes* que nous retenons dans notre ameublement du monde.

Par entité *complexe*, il faut entendre une entité qui « lie » ou « réticule » des entités *simples*. Un exemple d'entité complexe est celui de la *proposition*, que nous assimilons au sens ou au contenu mental d'énoncés tels « Paul construit une maison », « Paul aime Marie » ou « Paul a une température de 39°C ». De telles propositions sont communément considérées comme des touts constitués d'individus (ex : 'Paul', 'Marie', '39°C') et de relations (ex : 'construire', 'aimer', 'avoir pour température') [28][1]. Dans une théorie de la vérité fondée sur une correspondance entre propositions mentales et entités du monde, les propositions sont considérées comme étant *porteuses de vérité* (*truth-bearer*). La vérité des propositions est fondée sur l'existence d'entités du monde dites *vérifactrices* (*truth-maker*). Ainsi, c'est parce que dans le monde physique Paul a une température dont la magnitude est mesurée à 39°C que la proposition « Paul a une température de 39°C » est vraie. L'entité du monde physique que nous venons d'évoquer, à savoir le fait que Paul a une température de 39°C est communément assimilée à une entité complexe, appelée *état de choses*. Comme nous venons de le voir, nous avons donc affaire à des entités complexes mentales et physiques.

Pour rendre compte de la réticulation des entités complexes, nous faisons appel à la notion de *connexion*, une figure ontologique possédant un pedigree important mais qui a été éclipsée dans la métaphysique contemporaine par la théorie des relations de Bertrand Russell et Georges Moore[2]. Fondamentalement dans la littérature, la connexion est assimilée à une entité productrice de touts organisés – comme on le voit avec la notion de *nexus* chez Gustav Bergmann [3] –

[1] Selon un principe ontologique commun, auquel nous souscrivons, les touts et leurs constituants sont de même nature. Ainsi, il convient de considérer que les individus notés 'Paul', 'Marie' et '39°C', sont ici des entités mentales représentant des entités physiques. De même, comme nous le verrons par la suite, nous considérons que les relations sont de nature conceptuelle.

[2] Nous reprenons ici l'avis formulé par Frédéric Nef dans [22] *De la logique des relations à la métaphysique des connexions*. L'étude que nous menons dans l'article s'appuie sur les prémisses d'une métaphysique des connexions établies par Nef dans cet ouvrage.

et, en cela, la connexion diffère de la relation.

Pour notre étude, nous choisissons de nous focaliser en priorité sur les *états de choses* ou connexions physiques. Les choses en question sont des objets et des processus. Nous définissons à leur égard deux espèces de connexion, à savoir une connexion *interne* (ces substrats que sont les objets et processus sont connectés à leurs qualités) et une connexion *externe* (correspondant à des interactions entre objets et processus). Dans l'article, nous posons les bases de ces différentes connexions. Par ailleurs, nous soulignons que nous avons connaissance de types récurrents de connexions, qu'il s'agisse de l'*inhérence* des qualités à leur porteur ou de l'*énaction* de processus par un objet. Nous caractérisons ces *liens de connexion* généraux comme une espèce de *relation*.

Dans la suite de l'article, nous commençons par rappeler quels sont les présupposés ontologiques sur lesquels nous nous appuyons et notre notion d'*ontologie épistémique* (§ 2). Nous approfondissons ensuite la notion de qualité matérielle physique en ayant soin de distinguer, notamment en nous appuyant sur des données récentes de la psychologie de la perception, ce qui relève du mental (de notre connaissance du monde physique) et ce qui relève du physique (§ 3). À ce propos, l'*inhérence* de qualités à leur porteur est considérée comme un *modèle* du monde. Ceci nous conduit à préciser le type d'*état de choses* physique que nous retenons (§ 4). Enfin, nous définissons les liens généraux de connexion comme une espèce de *relation* (§ 5).

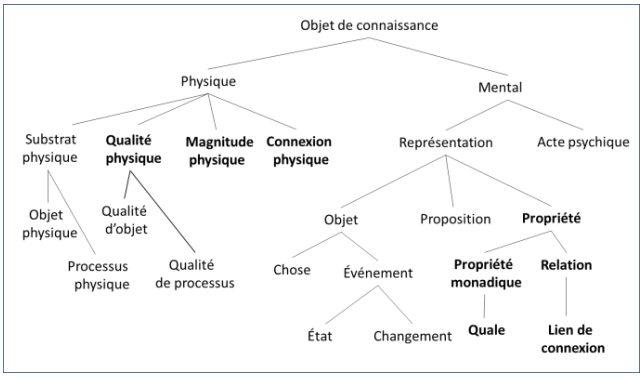


Fig. 1 : esquisse d'une ontologie fondatrice épistémique ; les catégories analysées dans l'article apparaissent en gras

2 Notre cadre ontologique de référence

Dans cette section, nous rappelons nos engagements ontologiques de base et la notion d'*ontologie épistémique* dont nous préconisons le développement [16]. Ces engagements sont fondés sur une théorie des objets intentionnels – une théorie de *l'objet de représentation* – développée au tournant du 20ème siècle dans l'école brentanienne par le philosophe et psychologue Kasimir Twardowski dans son manuscrit d'habilitation [31] *Sur la théorie du contenu et de l'objet des représentations*.

Selon cette théorie (*cf*. Fig. 2), lorsqu'un sujet pense à un objet, sa conscience se dirige vers un objet immanent (rel *IMM*). Lorsque le sujet conçoit cet objet comme transcendant à son esprit, il conçoit l'objet comme représentant une entité externe (rel *REPR*) à laquelle il se réfère (rel *REF*). Dans le cas où l'objet est conçu comme ne représentant pas d'entité transcendante, la visée s'arrête à l'objet immanent (les flèches en pointillés signifient que les entités correspondantes n'ont pas besoin d'exister). Une des motivations psychologiques de cette théorie est d'expliquer comment nous pouvons penser à des objets « non-existants », qu'il s'agisse d'objets impossibles (ex : 'le cercle carré'), d'objets jusqu'à présent jamais rencontrés (ex : 'une montagne d'or') ou d'objets physiques ayant cessé d'exister (ex : 'Aristote'). Sur un plan épistémique, les objets immanents constituent notre connaissance du monde présent, passé et possible. Pour souligner cette fonction, dans la suite de l'article, nous privilégions le terme « objet de connaissance ».

Il convient de noter que les représentations que nous venons d'évoquer correspondent à des représentations *abstraites* ou *conceptuelles* du monde. Lors d'une perception du monde physique, un sujet se construit des représentations sensorielles dont le contenu est d'un format différent, structuré au moyen d'entités pré-conceptuelles [25]. Nous les évoquerons en § 3. Ces représentations complètent notre connaissance du monde.

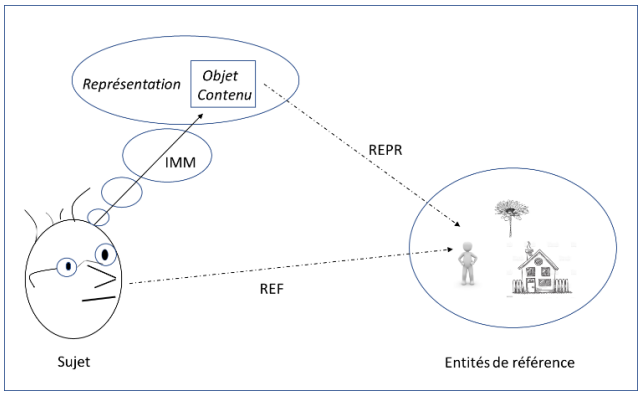


Fig. 2 : entités impliquées lors d'une pensée d'un sujet à un objet

Selon Twardowski, deux espèces d'objets de connaissance existent, resp. des objets *singuliers* et des objets *généraux*. Un objet singulier représente un individu doté de propriétés (ces dernières constituent le *contenu* de la représentation). Par exemple : l'objet singulier 'Simba' est représenté comme étant un jeune lion ; 'Mufasa' est un lion adulte. L'objet général, l'analogue mental d'une idée générale platonicienne, partage avec l'objet singulier le fait d'être une unité et non une pluralité. Sur un plan ontologique, l'objet général, par exemple 'le lion', 'l'hépatite' ou 'le non-fumeur', possède des déterminations communes à une pluralité d'objets singuliers. Sur le plan des représentations (*cf*. Fig. 3), et selon un principe d'économie, la représentation d'un objet général (ex : 'lion') est partagée par des représentations d'objets singuliers qui exemplifient l'objet général (ex : 'Simba' et 'Mufasa'). Sur un plan logique, l'objet général est l'objet de jugements spécifiques, par exemple le jugement selon lequel *le lion est carnivore*, *l'hépatite est une inflammation du foie* ou *le non-fumeur est moins sensible aux maladies cardio-pulmonaires que le fumeur*. De tels jugements ont une valeur (logique et épistémique) distincte de jugements portant sur des objets singuliers.

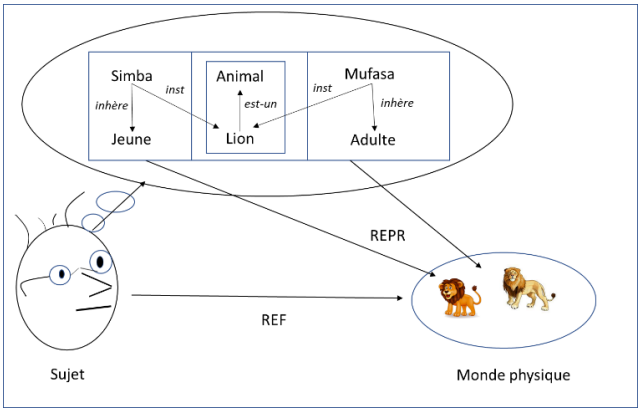


Fig. 3 : la représentation de l'objet général 'lion' est incluse dans les représentations des objets singuliers 'Simba' et 'Mufasa'

Dans [16], nous avons proposé d'adopter ce cadre ontologique pour définir une nouvelle espèce d'ontologie-artefact. Les catégories d'une ontologie *épistémique* correspondent à des objets généraux de connaissance. De ce fait, une ontologie épistémique vise à rendre compte de nos connaissances du monde plutôt que du monde directement. La catégorie racine `Objet de connaissance` d'une telle ontologie (*cf.* Fig. 1) représente ainsi un *quelque chose* en général auquel un sujet peut penser. Les catégories `Physique` et `Mental` représentent notre connaissance respectivement d'une entité physique et mentale en général. On notera l'adoption d'une bipartition des entités mondaines en entités physiques et mentales, une question continuant de faire débat en métaphysique contemporaine. Ce débat est couramment exprimé en termes de la distinction entre entités *concrètes* et *abstraites*. Les entités *concrètes* sont créditées d'une existence spatio-temporelle indépendante de toute pensée humaine. Il s'agit du critère que nous retenons pour nos entités physiques. Du côté *abstrait* sont communément distinguées, d'une part, des entités mentales et sociales dépendant d'esprits humains pour leur existence et, d'autre part, des entités idéales telles les entités mathématiques ayant une existence objective mais celle-ci n'étant ni spatiale ni temporelle. Notre catégorie d'entité mentale couvre ces deux dernières classes d'entités abstraites, à savoir des entités sociales comme une loi, un syndicat ou un anniversaire et des entités mathématiques comme les nombres et les figures géométriques.

Pour compléter nos présupposés ontologiques, rappelons un engagement important (notamment pour la théorie des propriétés / relations que nous adoptons, *cf.* § 5), à savoir un nominalisme des universaux. Un tel engagement signifie que nous réfutons l'existence d'universaux (physiques ou mentaux) pour plébisciter une ontologie de particuliers. Les universaux ont été historiquement introduits pour expliquer les ressemblances entre entités dans le monde. Ainsi, la ressemblance entre par exemple des êtres humains ou des moineaux est expliquée par le fait qu'un universel identique 'être humain' (resp. 'moineau') est présent dans chaque individu du même type[3]. La théorie des objets généraux de Twardowski fournit une explication différente : ce sont des objets mentaux qui rendent compte de la ressemblance entre entités du monde (et, comme nous l'avons rappelé, les représentations d'objets généraux sont partagées par les représentations d'objets singuliers). Bien sûr, cette ressemblance conçue mentalement a une contrepartie dans le monde physique mais un principe de parcimonie nous invite à ne pas introduire inutilement de nouvelle entité, surtout une entité aussi déroutante se répétant à l'identique dans de multiples objets[4].

Pour rappel, l'objet principal de notre étude dans cet article est de caractériser les états de choses physiques que nous voulons admettre dans notre ontologie. Nous avons évoqué en introduction une classe d'états de choses correspondant au fait qu'un objet « a » ou « porte » une « propriété » ou « qualité », par exemple au fait que *Paul a pour température 39°C*. Pour débuter notre étude, nous allons tout d'abord préciser notre notion de *propriété* ou *qualité*. Nous nous attèlerons ensuite à préciser ce que « avoir » ou « porter » une propriété / qualité signifie dans un cadre métaphysique privé d'universaux.

3 Qualités physiques et qualia phénoménaux

Dans cette section, nous définissons un cadre métaphysique pour les qualités physiques matérielles en assimilant l'*inhérence* de qualités particulières à leur porteur (objet ou processus) à un *modèle* de la réalité physique plutôt qu'à la réalité physique elle-même. Pour cela, nous nous livrons à une lecture cognitive des théories des « propriétés particulières » ou « tropes » telles qu'élaborées par ses pères fondateurs (notamment Donald D. Williams et Keith Campbell) et nous confrontons cette analyse à des données récentes des sciences de la couleur, en particulier de la psychologie de la perception, que nous extrapolons aux qualités perceptibles matérielles en général.

La philosophie des « propriétés particulières » ou « tropes » (pour reprendre le terme proposé par Williams et aujourd'hui largement consacré), s'est établie courant du 20ème siècle en défendant une structure ontologique nouvelle du monde (par rapport à la théorie aristotélicienne de la substance et ses accidents) : le monde est peuplé d'individus consistant en une collection de propriétés « compresentes » (pour reprendre cette fois le terme proposé par Bertrand Russell)[5]. Un point nous

[3] L'universel que nous venons de décrire est l'universel aristotélicien. Pour Platon, l'universel est une entité idéale flottant au-dessus des objets et leur étant préexistante.

[4] Une stratégie analogue en philosophie de la ressemblance est défendue actuellement par Gonzalo Rodriguez-Pereyra [26].

[5] Deux publications phare sont communément retenues par les théoriciens des tropes, à savoir l'article pionnier de Williams de 1953 *On the Elements of Being* (*cf.* [33] pour la traduction d'une première partie) et l'ouvrage de Campbell de 1990 *Abstract Particulars* (*cf.* [7] pour la traduction française d'un extrait traitant du problème des universaux). La notion de propriété particulière avait déjà fait parler d'elle dans un article de George F. Stout de 1921 *The Nature of Universals and Propositionss* (*cf.* [29] pour une traduction française). En réalité (historique), l'ontologie des tropes jouit d'un pedigree important puisqu'on peut faire remonter au Moyen Âge et à Pierre Abélard (1079-1142) l'évolution des accidents en tropes, comme le

interpelle tout particulièrement dans la littérature sur les tropes, à savoir leur qualification de particuliers « abstraits ». Nous y voyons une occasion de renouveler la question de la frontière entre le physique et le mental. En effet, un examen des textes fondateurs montre que l'abstraction en question est une activité cognitive, ce qui invite à rapprocher la qualité physique d'une représentation mentale.

Une « propriété particulière » est un constituant fondamental d'un individu : c'est *ce rouge* d'une rose, *cette dureté* d'un morceau de métal, *ce goût sucré* d'une sucette, *cette triangularité* d'un biscuit, *etc*. Ces propriétés se distinguent selon leur *nature* (une couleur est distincte d'une résistance au toucher, d'une saveur et d'une forme). Par ailleurs, des propriétés d'une même nature se ressemblent, entrainant des ressemblances entre individus les portant : le fait que deux individus distincts nous apparaissent comme rouges ou triangulaires s'explique ontologiquement par le fait qu'ils sont constitués de propriétés particulières de couleur et de forme qui se ressemblent, tout en étant numériquement distinctes.

Un point commun affirmé par les théoriciens des 'tropes' est qu'il s'agit *des objets immédiats de la perception*[6]. La perception, et plus largement nos processus psychiques, sont de fait conceptuellement associés à la nature ontique des tropes. L'*abstraction* est l'acte psychique par lequel nous appréhendons des aspects des corps matériels, comme le précise Campbell [7] :

> Un item est abstrait (…) s'il est amené devant l'esprit par un acte d'abstraction, c'est-à-dire par la concentration de l'attention sur quelque chose – non pas tout – de ce qui lui est présenté. Un corps matériel complet, une chaussure, un bateau, ou un morceau de cire à cacheter, sont concrets ; tout ce qui se trouve là où est la chaussure lui appartient – sa couleur, sa texture, sa composition chimique, sa température, son élasticité, et ainsi de suite : ce sont tous des aspects ou des éléments inclus dans l'être de la chaussure. Mais ces traits ou caractéristiques considérés de façon individuelle, par exemple la couleur de la chaussure ou sa texture, sont abstraits en comparaison d'autres.

Williams, avant Campbell, avait mis en avant deux activités psychiques – l'abstraction et la généralisation – à l'œuvre dans notre façon d'appréhender le monde [*ibid.*, p. 176] :

> Parmi les nombreux processus appelés « abstraction », seul le plus primitif mérite ce nom : la conscience distincte de l'abstractum lui-même, qui se produit au niveau sensoriel et même au niveau animal. À peine plus élevée est une généralisation rudimentaire, la propension à traiter de façon similaire des abstracta similaires ; mais les offices de la conception sont nécessaires pour prendre conscience soit qu'un abstractum donné est abstrait (et appartient à une somme de concurrence), soit qu'il exemplifie un universel (et appartient à un ensemble de similitude).

La citation de Williams est intéressante car elle évoque les « offices » de nos conceptions dans ce qui s'avère être un processus de construction d'un modèle du monde distinguant des constituants abstraits, des individus concrets et des ensembles ou classes de ressemblance de ces entités abstraites et concrètes. Des thèses récentes en psychologie de la perception confirment cette activité de construction d'un modèle du monde physique. Ainsi, selon le psychologue Rainer Mausfeld, la théorie assimilant la perception à la « redécouverte » ou « reconstruction » d'une scène extérieure est tout simplement erronée [20]. Cette thèse, toujours selon Mausfeld, se fonde sur une métaphysique réaliste – trop – naïve selon laquelle les entités du monde externe sont le déclaque de nos percepts. La perception consiste de fait en l'élaboration de modèles sémantiques à partir d'inputs sensoriels proximaux consistant en patterns d'énergie spatio-temporels physiques. Nous tirons de ces données psychologiques deux conséquences.

D'une part, cette construction de classes de ressemblances est en cohérence avec notre nominalisme des universaux. Il suffit d'identifier ces classes de ressemblances aux objets généraux de connaissances que nous évoquions en § 2. Que des individus physiques et leurs qualités soient de nature proche ne nécessite pas pour autant que des classes de ressemblance existent physiquement.

D'autre part, ce même processus constructif milite en faveur d'assimiler les individus particuliers constitués de propriétés particulières inhérentes, et donc les propriétés particulières elles-mêmes, à un *modèle* du monde, autrement dit à nos connaissances du monde plutôt que de relever de la réalité physique ultime.

Dans les études multidisciplinaires (physique et psychologique) contemporaines appliquées à la couleur, nous retrouvons convoqués les niveaux physique et mental. Ainsi, Alex Byrne et David Hilbert, dans leur article de référence [6] *Color realism and color science*, formulent la proposition suivante. Lors d'une expérience véridique de vision d'un objet, deux « couleurs » sont à distinguer :

(i) Côté physique, l'objet possède une propriété - la *couleur physique* - intervenant causalement dans un phénomène de réflectance de la lumière illuminant la surface de l'objet ;

(ii) Côté mental, le sujet est dans un état représentationnel ayant pour contenu la proposition 'l'objet est rouge', cette proposition correspondant à la *couleur phénoménale*.

Faisons le lien avec la notion de qualité matérielle telle que nous venons de la définir *supra*, c'est-à-dire entendue mentalement comme connaissance du monde physique. Le cadre métaphysique que nous retenons est illustré en Fig. 4.

Ce que B&H désignent par « couleur physique » correspond à un phénomène physique qu'un sujet se représente comme une qualité. En Fig. 4, l'entité du monde physique `PP-Couleur-Rouge`$_{\#1}$ tient pour un phénomène physique (PP) particulier de rougeur que se représente un sujet au moyen de l'objet mental `Couleur`$_{\#}$, ce dernier tenant pour une qualité particulière. Précisément, B&H définissent la couleur physique comme : « la proportion de lumière incidente que l'objet est disposé à réfléchir à chaque longueur d'onde du spectre visible » [*ibid.*, p. 9]. On peut noter que cette définition – qui caractérise la

montre l'étude d'Alain de Libéra [17].

[6] Ceci ne vaut bien sûr que pour les tropes sensibles et non des tropes comme la masse d'un corps matériel ou sa concentration en un élément chimique, lesquels peuvent faire l'objet d'un mesurage.

couleur comme indépendante de toute illumination contextuelle et ainsi comme persistante dans l'objet – est cohérente avec la notion *de sens commun* de couleur. Cette notion de propriété dispositionnelle faisant appel à la réflectance de la lumière selon une longueur d'onde particulière est simplement une notion *savante* de qualité de couleur.

Quant à la « couleur phénoménale », elle correspond à la notion de *quale* (*qualia* au pluriel) utilisée par les philosophes pour faire référence aux aspects phénoménaux de notre vie mentale[7]. Suivant notre ontologie des représentations conceptuelles (*cf.* Fig. 4), un quale (ex : $\texttt{Quale}_{\#k}$) a – en tant que représentation – pour objet une qualité de couleur (ex : $\texttt{Couleur}_{\#i}$) caractérisée par plusieurs propriétés (ex : $\texttt{Couleur}_{\#i}$ est inhérente dans $\texttt{Objet}_{\#j}$, est de type `Couleur` et a pour magnitude `Rouge`).

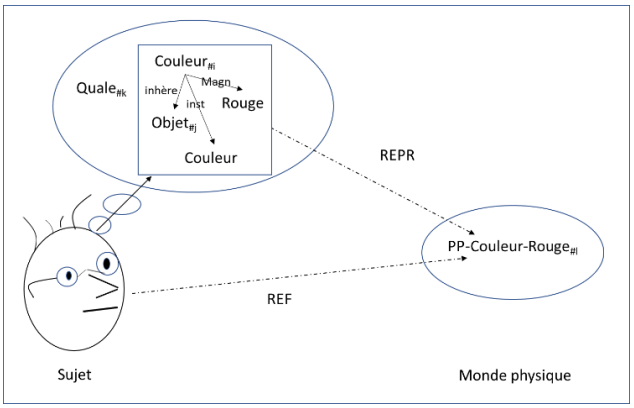


Fig. 4 : la qualité particulière $\texttt{Rouge}_{\#i}$ représente pour le sujet un phénomène physique particulier $\texttt{PP-Rouge}_{\#j}$

Pour compléter le cadre métaphysique venant d'être esquissé concernant les qualités matérielles d'objets physiques, nous ajoutons deux remarques.

Du côté du monde physique, nous avons évoqué un phénomène physique de couleur causalement responsable d'un quale de couleur. Notons qu'une telle analyse s'étend aux qualités matérielles en général. La connaissance scientifique permet en effet d'identifier pour chaque qualité un phénomène physique qui en soit à l'origine. De tels phénomènes impliquent la structure intime de l'objet : la *température* d'un corps dépend de l'agitation de ses molécules ou atomes ; la *fluidité* de l'eau dépend de l'organisation géométrique de la molécule d'eau associée à des molécules d'autres éléments ; l'*odeur* d'un objet dépend de centaines de milliers de particules émanant de l'objet et venant se fixer sur des récepteurs de nos muqueuses nasales.

Du côté de nos connaissances de sens commun du monde physique, le modèle des qualités est couramment étendu dans plusieurs directions. Il est notamment habituel de reconnaître des qualités *complexes* constituées de qualités *simples*, à l'instar d'une *couleur* constituée d'une *teinte*, une *brillance* et une *saturation*. La justification donnée est que de telles qualités primitives représentent des phénomènes physiques distincts : une *longueur d'onde* pour la teinte, une *luminance* pour la brillance, enfin une *pureté* pour la saturation. Par ailleurs, comme le propose Nicola Guarino [12], des applications peuvent nécessiter de considérer des qualités *globales* et *locales* d'un objet[8], voire de détailler davantage en considérant des champs de qualités. De telles extensions correspondent à des niveaux de connaissances plus fins que nous avons du monde physique.

Dans la suite de l'article, compte tenu de notre propos, nous nous en tenons au modèle simplifié du monde de la Fig. 1. Nous nous apprêtons à admettre la qualité-phénomène physique comme constituant d'états de choses. Quant au quale-concept, nous allons l'admettre comme constituant de représentations, telle la proposition.

4 Connexions physiques

Dans cette section, nous précisons le type de fait physique ou *état de choses* que nous retenons dans notre ontologie[9]. Dans la littérature philosophique, deux principales théories d'états de choses ont été définies se distinguant suivant le rôle (sémantique, métaphysique) attribué à ces entités et, par là-même, suivant l'argument mis en avant pour justifier leur existence.

Une première catégorie d'états de choses a été introduite sur un plan sémantique comme *véri-facteur* (*truth-maker*) de propositions (rappelons que ces dernières sont porteuses de vérité). Ainsi, pour expliquer la véracité de propositions telles 'Paul est triste', 'Paul aime Marie' ou 'Paul échange des secrets d'états avec Marie', il est postulé l'existence des états de choses suivants :

<Paul, triste>

<Paul, aime, Marie>

<Paul, échange des secrets d'état avec, Marie>

De tels états de choses sont constitués d'un (ou plusieurs) particuliers (ex : 'Paul', 'Marie') et d'une propriété (ex : 'être triste') ou de relations (ex : 'aimer', 'partager des secrets d'état avec'). Le philosophe David Armstrong est le plus illustre défenseur contemporain d'une telle stratégie abductive de la

[7] Par « aspect phénoménal », il faut comprendre que des couleurs, saveurs ou odeurs particulières occasionnent des effets divers sur notre vie psychique. Par exemple, sentir une odeur de rose ou une odeur d'œufs pourris ne provoque pas les mêmes effets. Dans [1], Liliana Albertazzi et Roberto Poli mentionnent différentes dimensions phénoménales de la couleur auxquelles nous faisons référence au moyen d'expressions comme « couleur chaude » *vs.* « couleur froide », ou « couleur pâle » *vs.* « couleur intense ». Les théories philosophiques divergent quant au statut ontologique à accorder aux qualia. Dans cet article, nous leur conférons le statut d'objet de représentation. Fred Dreske [10] est connu pour avoir également proposé une théorie représentationnelle de l'expérience perceptuelle au moyen de qualia.

[8] Un exemple que donne Guarino [*ibid.*] est celui d'un fleuve porteur d'une *longueur globale* et de différentes *largeurs locales* selon le tronçon du fleuve considéré.

[9] Le terme français « état de choses » est la traduction du terme anglais « state of affairs » et du terme allemand plus explicite « sachverhalt » (littéralement « comportement » [verhalt] de « chose » [sach] »).

meilleure explication à donner de la véracité des propositions [2]. Cependant, comme le soutient Arianna Betti, postuler l'existence de telles entités pour une raison sémantique et justifier leur existence sur un plan ontologique sont deux choses différentes [4] :

> Les travaux récents en métaphysique analytique expriment toutefois une insatisfaction croissante par rapport aux arguments *a priori* de cette espèce en faveur d'entités ou contre elles, en particulier par rapport aux arguments sémantique *a priori*. Si on veut argumenter en faveur d'une certaine catégorie ontologique, suivant cette récente tendance, il faut de véritables arguments ontologiques.

De fait, plusieurs arguments jouent en défaveur de tels états de choses. D'une part, le fait d'admettre un (ou plusieurs) constituant(s) particuliers et un constituant universel, autrement dit le fait que le complexe soit non homogène, les classe parmi les chimères ontologiques. Rappelons que, pour ce qui nous nous concerne, nous avons adopté comme présupposé ontologique un nominalisme des universaux. D'autre part, à notre connaissance, aucune théorie satisfaisante n'a permis à ce jour d'expliquer la nature de la « glue » ou du « ciment » permettant aux *reticula* – le(s) particulier(s) et l'universel – de constituer un tout[10].

En l'occurrence, suivant notre cadre ontologique et pour rester dans le cadre d'une théorie de la correspondance de la vérité, nous proposons d'autres véri-facteurs pour les propositions données à titre d'exemples. Dans le cas des propositions 'Paul est triste' et 'Paul aime Marie', nous identifions un état affectif de Paul, respectivement négatif et positif, et dirigé vers Marie pour la seconde proposition. Ce sont là des entités mentales. Dans le cas de la proposition 'Paul échange des secrets d'état avec Marie', nous lui attribuons plusieurs véri-facteurs correspondant à des actions réalisées par Paul et Marie pour s'échanger ces fameux secrets. Parmi ces véri-facteurs figurent des entités mentales (ex : des intentions d'agir) et des entités physiques, dont des états de choses, mais ces derniers étant à entendre dans un sens plus contraint que nous nous apprêtons à définir.

Une seconde catégorie d'états de choses a été introduite par Bertrand Russell dans son explication du mouvement continu, qu'il définit ainsi [27, chap. 54], « Le mouvement consiste simplement à occuper des lieux différents à des moments différents ». Ainsi, selon Russell, un mouvement continu d'un objet *O* n'est rien d'autre (ni plus, ni moins) qu'une série de faits correspondant à l'occupation *Loc* par *O* de différentes positions Pos_i à des instants successifs I_j :

<O, Loc, Pos_1, I_1>, <O, Loc, Pos_2, I_2>, …

Cette théorie est nommée en anglais 'at-at', les faits instantanés signifiant qu'un objet se situe *à* (*at*) différentes positions *à* (*at*) différents instants. Une telle théorie peut être avancée pour rendre compte du changement temporel en général, et pas uniquement du mouvement, notamment du changement de qualité pour un objet…

<Paul, Inhère, $Température_{Paul\_1}$, I_1>, <Paul, Inhère, $Température_{Paul\_2}$, I_2>, …[11]

… ou un processus (ci-dessous, $Marche_{Paul}$ représente une instance de processus de marche de Paul) :

<$Marche_{Paul}$, Inhère, $Vitesse_{MarchePaul\_1}$, I_1>, <$Marche_{Paul}$, Inhère, $Vitesse_{MarchePaul\_2}$, I_2>, …

Qu'en est-il de la validité de la théorie 'at-at' ? Dès le début du 20ème siècle, celle-ci a fait l'objet de critiques au prétexte qu'en assimilant le mouvement à une série d'immobilités (des états) elle ne permet pas de rendre compte de la dynamique du mouvement. Par ailleurs, comme l'a souligné ultérieurement Peter Geach [11], cette théorie ne permet pas non plus de distinguer les « vrais » changements émanant des objets changeants (et nécessitant de leur part une dépense d'énergie) de simples « changements de Cambridge »[12]. De nos jours, pour caractériser les vrais changements, la théorie 'at-at' est complétée par l'introduction dans l'ameublement ontologique d'une entité permettant d'expliquer qu'un objet passe d'un état à l'autre, à savoir le *processus*[13]. Le fait que l'objet se mouvant *énacte* un processus tel $Marche_{Paul}$ permet de rendre compte à la fois de la dynamique du mouvement et de la responsabilité causale (en termes de dépense d'énergie) de l'objet se mouvant. La série de faits rend compte pour sa part d'une évolution du monde : pour qu'un objet passe d'une position à une autre, il est nécessaire qu'il passe par des positions intermédiaires ; de plus, pour être dans la position Pos_{In} à l'instant I_n, autrement dit pour que l'état de choses <O, Loc, Pos_{In}, I_n> tienne, il est nécessaire que l'état de choses <O, Loc, Pos_{In-1}, I_{n-1}> ne tienne plus.

La destinée des faits instantanés de la théorie 'at-at' semble être différente de celle des états de choses armstrongiens (évoquée *supra*) dans la mesure où l'argument de leur existence est de nature métaphysique, précisément physique, et non sémantique[14]. Dans le même temps, toutefois, tels que définis,

[10] Pour un exposé très clair et convaincant de l'échec des pistes envisagées, le lecteur peut se référer à Betti [4] qui conclut son article ainsi (p. 250) : « Si les objections présentées ci-dessus sont justes, alors il faut dire adieu aux théories des faits qui réticulent des universaux ». En d'autres termes, les seuls touts admissibles sur un plan ontologique sont des touts homogènes dont tous les constituants sont de même nature (ce que nous rappelions dans notre introduction).

[11] Pour être plus précis, nous devrions parler du changement de magnitude de qualités avec des états de choses prenant comme objets les qualités, par exemple : <$Température_{Paul}$, a pour valeur, 37,9 °C, Ij>. Par la suite, à titre de simplification, nous continuerons d'utiliser la notation abrégée <Paul, Inhère, Température de 39°C$_{Paul}$, Ij>.

[12] Le terme « changement de Cambridge » a été proposé par Peter Geach [*ibid.*, p. 13] pour dénoter la conception du changement continu promue par des philosophes de Cambridge dont John McTaggart et Bertrand Russell. Comme exemple de changement de Cambridge se résumant à la série de faits précitée sans nécessiter de dépense d'énergie de la part d'un objet, on peut citer le transport d'un objet.

[13] Comme proposé notamment par Caroll Cleland [9]. Nous avons exposé dans [14] et [15] notre adoption de la conception du processus de Cleland.

[14] Rappelons que nous ne nous intéressons dans cette section qu'aux seuls états de choses *physiques* existant indépendamment de toute pensée humaine. Nous laissons ainsi de côté des états de choses tel 'le prix de la baguette est de 1€20' qui est un fait social [30]. En

ils semblent avoir pour constituants des particuliers et des universaux (par exemple, les relations 'Loc' et 'Inhère') et devraient donc être écartés au motif d'être des complexes inhomogènes, donc des chimères ontologiques. Si nous voulons défendre leur existence, il est nécessaire, pour reprendre l'avis précité de Betti d'avancer « de véritables arguments ontologiques ».

Pour ce faire, nous allons reprendre la théorie des qualités matérielles esquissée en § 3 distinguant des qualités conceptuelles et des qualités-phénomènes physiques, que nous allons compléter en nous appuyant sur une théorie de la perception que nous exposons maintenant.

Des avancées récentes concernant les bases neurologiques de la perception indiquent que les humains échantillonnent les données du monde physique à une fréquence d'environ 13 instantanés par seconde (cette fréquence variant selon les individus) et que nos systèmes perceptifs / cognitifs reconstituent et présentent à la conscience des états ou changements continus[15]. Le mécanisme à l'œuvre est similaire au procédé cinématographique exploitant 24 images par seconde pour nous donner l'impression d'un mouvement continu à partir d'images fixes. Ce mécanisme, fondé sur des prédictions (le système perceptuel remplit les trous sur la base de prédictions), joue le rôle d'un cinéma intérieur pour donner à la conscience l'illusion d'un état ou d'un mouvement continu[16].

Sur la base de ces données, nous proposons un cadre métaphysique de la perception faisant intervenir trois types d'entités (*cf.* Fig. 5).

Côté physique existent des états de choses correspondant au fait, pour des objets (et processus), d'être le siège de phénomènes physiques particuliers. Ces phénomènes sont des réalisations instantanées de propriétés dispositionnelles, telles que définies par Byrne et Hilbert (*cf.* [6] et notre présentation en § 3). Par exemple (*cf.* Fig. 5), l'état de choses [Paul, PP-Température$_{\#j}$, I$_{\#j}$] correspond au fait que 'Paul' soit le siège du phénomène physique de température 'PP-Température$_{\#j}$' à l'instant 'I$_{\#j}$' . Nous nommons ces états de choses « connexions physiques ». Selon la nature de ces états de choses, aucune entité supplémentaire de type « glue » ou « ciment » n'est à prendre en compte pour justifier l'existence du complexe, ce qui correspond à la notion de *nexus* chez Bergman [3][17]. Ces connexions physiques sont les entités immédiates de la perception (en tout cas pour certaines d'entre elles).

Côté mental, à un niveau pré-conceptuel, des représentations équivalentes aux faits instantanés de la théorie 'at-at' sont causalement créées lors d'une perception, par exemple : <Paul, dans, Température$_{\#j}$, I$_{\#j}$> (le constituant 'dans' correspond à l'information pré-conceptuelle de la présence de 'Température$_{\#j}$' dans l'objet 'Paul' ; l'instant I$_{\#i}$ dépend de l'échantillonnage évoqué *supra*). Rappelons que nous supposons que lors d'une perception, plusieurs représentations de formats différents sont construites. Nous avons évoqué en § 2 des représentations abstraites conceptuelles et des représentations intuitives sensorielles. La question du la structuration du contenu de ces représentations continue de faire débat [25].

De telles représentations sensorielles, résultats de mécanismes sub-personnels (c'est-à-dire inconscients), conduisent à la présentation à la conscience d'*états conceptuels* désignés par des termes tels 'l'avoir par Paul d'une température de 39°C' ou 'l'avoir par la marche de Paul d'une vitesse de 6 km/h'[18]. De tels états correspondent au contenu de phrases telles « Paul a pour température 39°C » ou « Paul marche à la vitesse de 6 km/h ». Nous proposons qu'ils aient comme structure, respectivement :

<Paul, Inhère, Température de 39°C$_{Paul}$, maintenant>

<Marche$_{Paul}$, Inhère, Vitesse de 6 km/h$_{MarchePaul}$, maintenant>

Ainsi côté mental, et cette fois au niveau conceptuel, les constituants des états sont tous des entités mentales et conceptuelles. Les entités 'Température de 39°C$_{Paul}$' et 'Vitesse de 3 km/h$_{MarchePaul}$' sont des qualités, telles que définies en § 3. Ces qualités représentent (*cf.* Fig. 2, rel *REPR*) des qualités-phénomènes physiques spécifiques et particuliers. La relation d'*inhérence* représente le fait que le phénomène physique est littéralement *dans* l'objet, ce qui correspond à l'étymologie du terme. Nous touchons là une limite en termes de résolution de notre connaissance des phénomènes physiques : nous ne connaissons pas précisément la nature des phénomènes en jeu (nous parlons là d'une connaissance de sens commun[19]). Enfin, le constituant temporel 'Maintenant' correspond à un temps subjectif.

revanche, nous nous intéressons à caractériser l'état de choses 'le poids de cette baguette est de 150 grammes' que l'on peut *a priori* (mais nous allons revenir sur cet *a priori*) assimiler à un fait physique.

[15] Concernant la perception visuelle, ce phénomène, déjà imaginé dans les années 90, a reçu une explication probante fondée sur des études expérimentales couplées à de la neuro-imagerie, comme présenté dans la publication de Rufin vanRullen *et. al.* [32].

[16] Comme le montre Lionel Naccache dans son [21] *Le cinéma intérieur*. Sur la base d'une fréquence d'échantillonnage de 13 Hertz, Naccache explique pourquoi, en visionnant dans un western le déplacement rapide d'une diligence ou en percevant directement dans la rue le déplacement rapide d'une voiture dans les jantes sur les roues portent des motifs, nous voyons littéralement les roues tourner en sens inverse.

[17] Pour une présentation synthétique de la notion de *nexus* chez Bergmann, le lecteur peut se référer à l'étude de Nef [22].

[18] Dans [15], nous considérons également des *changements* conceptuels (une autre espèce d'événement à côté des états) désignés par des termes tels 'l'augmentation de la température de Paul' et 'l'accroissement de la vitesse de marche de Paul'. Pour des raisons de simplification, nous n'évoquons dans le présent article que des états.

[19] En § 3, nous avons identifié les qualités-phénomènes physiques à des phénomènes impliquant la structure intime des objets comme, par exemple dans le cas d'une odeur, au fait que des centaines de milliers de particules émanant d'un objet se fixent sur les muqueuses nasales de la personne sentant. Il convient de noter qu'il s'agit là d'une connaissance scientifique et non plus de sens commun du phénomène. Dans une ontologie épistémique, compte tenu de la dimension conceptuelle des catégories, si nous voulions rendre compte de cette connaissance scientifique, nous aurions à considérer deux objets-qualités distincts, représentations distinctes d'une même réalité.

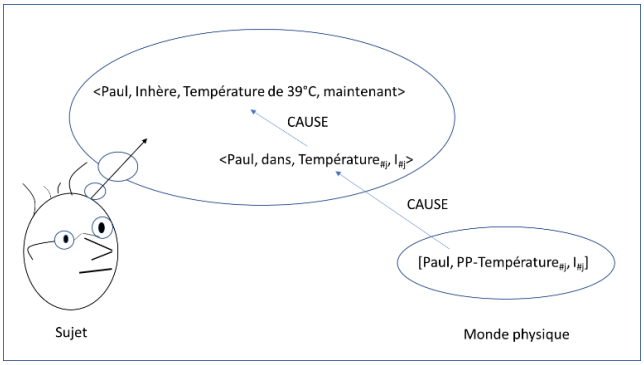


Fig. 5 : Entités impliquées lors d'une perception du monde physique

En conclusion, nous venons d'admettre dans notre ameublement du monde des états de choses ou connexions physiques correspondant au fait que des objets et des processus soient le siège de phénomènes physiques, de telles connexions étant des objets immédiats de perception. Ces connexions ayant une localisation spatio-temporelle unique, elles sont des entités particulières. Par ailleurs, nous avons montré que nous avons *in fine* une connaissance abstraite de ces états de choses prenant la forme d'états conceptuels ayant pour constituants des relations telles 'Loc' et 'Inhère'. Dans le section suivante, nous donnons une caractérisation de ces « liens de connexion » comme une espèce de relation.

5 Liens de connexion et relations

Dans cette section, nous caractérisons les liens de connexion comme (i) représentant directement des états de choses et (ii) comme une espèce de relation dont les relata représentent des entités nécessairement présentes à chaque instant où l'état de choses ou la connexion tient. À noter que, sur un plan terminologique, nous nommons « lien de connexion » un *type* de lien, autrement dit quelque chose dont nous savons qu'il se répète, ce qui justifie de considérer ces liens de connexion comme une espèce de relation (*cf.* Fig. 1). Les « liens de connexions » sont donc à distinguer des « connexions physiques », lesquelles réalisent physiquement une connexion particulière.

La double caractérisation des liens de connexion – (i) et (ii) – nous conduit à évoquer d'autres exemples de connexions entre objets et processus. En préambule, nous rappelons que nous assimilons les propriétés / relations à des entités mentales.

Comme communément définies, les propriétés / relations correspondent à « ce qui est dit » d'entités et sont exprimées en langue par un prédicat. Selon cette définition, les relations sont de même nature que les propriétés. Il s'agit d'une espèce de propriétés impliquant dans ce qui est dit plusieurs entités (leur *arité* est strictement supérieure à 1). Il convient toutefois de noter que leur *nature* (concrète ou abstraite) continue d'alimenter des débats en métaphysique contemporaine[20].

Sur un plan historique, rappelons que pour la majorité des scolastiques du Moyen Âge, les propriétés / relations sont mentales, autrement dit n'existent pas physiquement[21]. À notre époque, une des raisons de leur rejet (toujours en tant qu'entités physiques) tient à la difficulté de les localiser. Considérons l'information selon laquelle 'Paris est à l'ouest de Strasbourg'. La relation 'être à l'ouest de' nous renseigne sur la localisation relative des deux villes. Par contre, à supposer que la relation soit physique, dans quelle région spatio-temporelle existe-t-elle ? Une réponse négative consiste à dire que la relation n'est localisée ni à la position occupée par Paris, ni à la position occupée par Strasbourg. Il semble qu'une réponse positive nécessiterait de modifier la notion de localisation spatio-temporelle utilisée pour les objets. De fait, une raison principale du rejet des relations physiques tient à leur nature « étrange ». Considérons l'information selon laquelle 'Tom est plus grand que Sibylle'. La relation comparative 'être plus grand que' nous renseigne sur les tailles respectives de deux personnes. Dès lors, est-il raisonnable de considérer que de telles relations existent physiquement ? Le cas échéant, nous devrions le faire pour chaque couple de tropes de même nature (comparables) inhérents dans des entités physiques différentes, conduisant à considérer un nombre incroyablement élevé de relations pour un usage indéterminé[22]. Nous serions dès lors en porte à faux avec un principe de parcimonie prévalent communément en métaphysique. Dans ces conditions, nous optons pour une nature conceptuelle informationnelle de la relation[23]. Nous assimilons les propriétés / relations à des objets généraux de connaissance mentaux (*cf.* Fig. 1).

Ceci étant rappelé, nous évoquons une propriété caractéristique des *liens de connexion*, à savoir la présence

[20] Le constat peut être fait à la lecture des entrées 'Properties' et 'Relations' de l'Encyclopédie philosophique de Stanford (resp. [24] et [19]).

[21] Comme le montre l'étude de Jeffrey Brower [5]. De fait, si les relations n'existent pas pour Guillaume Occam (1285-1347), leur existence était déjà fortement questionnée par Thomas d'Aquin (1225-1274) et Jean Duns Scot (1266-1308), comme nous le rappelle Nef [23, p. 47] (citant lui-même des citations de ces scolastiques extraites de l'ouvrage de Massimo Mugnai *Leibniz's Theory of Relations*): « Duns Scot et Thomas s'accordent eux sur la débilité ontologique des relations sans affirmer leur non existence radicale : « La relation (*relatio*) est le plus faible (*debilissum*) de tous les êtres, puisqu'il n'est qu'un rapport (*habitudo*) entre deux autres choses et n'est que très peu connaissable par soi-même ». « La relation a un être très faible (*debilissimum*) ». ».

[22] Rappelons que, selon Armstrong [2], de telles relations « internes », dépendant de propriétés monadiques intrinsèques de leur porteur, n'apportent rien sur le plan ontologique – suivant son expression, elles consistent en un « déjeuner gratuit » (un « free lunch »). Récemment, le philosophe Jonathan Lowe, généralisant son propos à tout type de relation physique (y compris causale), a recommandé de ne pas consacrer de temps à essayer de démystifier cette notion étrange de relation « réelle » (« physique ») [18] « L'idée même de propriétés polyadiques est hautement mystérieuse et donne lieu à un ensemble de problèmes philosophiques. Si les arguments dans ce chapitre sont corrects, alors considérer de telles propriétés ne sert aucun objectif ontologique utile, aussi nous ne devrions pas passer trop de temps à essayer de les démystifier ».

[23] Nef [22], pour sa part, adopte ce parti pris pour sa théorie des relations.

nécessaire des relata représentés lorsque la connexion tient.

Comme nous l'indiquions en § 4, les seuls états de choses physiques dont nous admettons l'existence sont ceux pour lesquels la relation représente un phénomène physique élémentaire spécifique (non répétable). En cohérence avec cette caractérisation, nous posons qu'il est nécessaire que les *relata* d'un lien de connexion représentent des entités présentes à chaque instant où la connexion tient. À chaque instant où un objet est *localisé* dans une position, l'objet et la position existent concomitamment. De même, à chaque instant où une couleur *inhère* dans un objet, ou bien une vitesse *inhère* dans un processus, les objets, processus et leurs qualités existent concomitamment.

Sur la base de cette caractérisation, nous pouvons mentionner d'autres exemples de connexions impliquant cette fois des processus. L'énaction par un objet d'un processus peut être assimilée à une connexion, comme dans l'exemple :

<Paul, Énacte, $\text{Marche}_{\text{Paul}}$, Maintenant>

Un exemple de connexion liant des processus entre eux est la perpétuation causale de processus, autrement dit le fait que l'existence d'un processus soit maintenue causalement par un autre processus via le contact d'objets, ce qui intervient quand une personne pousse un objet. Par exemple, Paul, en énactant un processus de poussée, déplace une table.

<$\text{Poussée}_{\text{Paul}}$, Perpétue, $\text{Processus de déplacement}_{\text{Table}}$, maintenant>

Les liens de connexion 'Enacte' et 'Perpétue' représentent bien un processus physique spécifique et leurs relata (les entités représentées) sont nécessairement présentes concomitamment.

Examinons maintenant des relations qui ne sont pas des connexions. Pour rester dans la sphère physique, nous pouvons citer des relations spatiales 'être à côté' ou 'être au-dessus' et des relations comparatives telle 'être plus grand' ou 'être plus lourd'. Deux différences apparaissent.

D'une part, ces relations ne représentent pas d'uniques phénomènes physiques[24]. En d'autres termes, l'occurrence d'états relationnels n'est pas déterminée par un unique vérifacteur. Ainsi, les vérifacteurs de l'état de proximité spatiale 'Paul est à côté de Marie' sont les états de localisation resp. de Paul et de Marie, auxquels il faut ajouter la distance séparant leur position. De même, des états de comparaison de qualités admettent comme vérifacteurs les états de qualité des deux *relata*.

Par ailleurs, les états relationnels ne supposent pas, pour être vrais, la présence simultanée des relata (des entités représentées). A titre d'illustration, on peut considérer les exemples suivants : 'De gaulle était plus grand que César' ; 'Churchill a vécu plus longtemps que Aristote'. De fait, comme ces exemples le montrent, les relations en général étendent notre domaine de connaissance en permettant de considérer des événements impliquant des entités n'existant pas simultanément.

6 Conclusion

En distinguant dans cet article *connexions* et *relations*, nous avons convoqué deux ordres ontologiques distincts. D'un côté, nous avons assimilé les connexions à des productions de touts (de complexes), ces derniers pouvant être physiques et mentaux. D'un autre côté, nous avons assimilé les relations à des entités mentales conceptuelles, constituants de nos connaissances du monde. Un lien entre ces deux ordres ontologiques réside dans les *liens de connexion*, une espèce de relation correspondant à notre connaissance des connexions.

Dans notre étude des connexions, nous avons privilégié les connexions physiques, ou états de choses, et identifié deux espèces, à savoir les connexions « internes » de qualités physiques à leur porteur (objet ou processus) et les connexions « externes » entre objets et processus. La question se pose de savoir si nous avons été exhaustifs dans notre recensement des types de connexions physiques, sachant que nous avons procédé essentiellement par la monstration d'exemples ? Un premier élément de réponse consiste à noter que les objets physiques sont en eux-mêmes des touts connectés. Simplement la nature de leur connexion au niveau de la matière échappe à notre connaissance de sens commun. Nef [23], dans sa recension, évoque des connexions entre objets matérialisées par de nouveaux objets physiques (ex : des clous, des boulons) ainsi que des imbrications d'objets physiques (à l'instar de briques de légo) – nous sommes dans la production de nouveaux touts. Un élément complémentaire de réponse revient à noter qu'il existe également des connexions entre objets physiques s'exerçant à distance, comme celles correspondant aux forces de l'attraction et de la gravitation. Ceci nous indique que le chantier de l'étude des connexions que nous avons ouvert reste à poursuivre.

Références

[1] L. Albertazzi & R. Poli, Multi-leveled objects: color as a case study, *Frontiers in Psychology*, Vol. 5, Article 592, 2014 ; https://psycnet.apa.org/doi/10.3389/fpsyg.2014.00592

[2] D.M. Armstrong, *A world of states of Affairs*, Cambridge University Press, 1997.

[3] G. Bergmann, *Realism. A Critique of Brentano and Meinong*, Univ of Wisconsin, 1967.

[4] A. Betti, Contre les faits, dans J. Benoist (ed.), *Propositions et états de choses. Entre être et sens* (pp. 231-250), Paris, Vrin, 2006.

[5] J. Brower, Medieval Theories of Relations, in E.N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy (Winter 2018 Edition)*, URL = https://plato.stanford.edu/archives/win2018/entries/relations-medieval/

[6] A. Byrne & D.R. Hilbert, Color realism and color science, *Behavioral and Brain Science*, Vol. 26, N° 1, pp. 3-21, 2003.

[24] Pour mémoire, nous formulions cette même remarque au § 4 à propos de la relation 'partager des secrets d'états'.

[7] K. Campbell, Le problème des universaux, dans Cl. Panaccio (dir.), *Le nominalisme. Ontologie, langage et connaissance*, Paris (pp. 125-146), Vrin, 2012 ; trad. de A.-M. Boisvert & Cl. Panaccio de *Abstract Particulars* (pp. 27-45), Oxford, Blackwell, 1990.

[8] K. Campbell, La métaphysique des particuliers abstraits. http://semaihp.blogspot.fr/2014/08/traduction-de-keith-campbell-metaphysic.html (2014) ; trad. par M. Jeddi et B. Langlet de The Metaphysics of Abstract Particulars, in P. French (ed.), *Midwest Studies in Philosophy VI: The foundations of Analytic philosophy* (pp. 477-88), University of Minnesota, 1981.

[9] C.E. Cleland, The Difference Between Real Change and Mere Cambridge Change, *Philosophical Studies*, Vol. 60, pp. 257-280, 1990.

[10] F. Dretske, *Naturalizing the Mind – The Jean Nicod Lectures – 1994 Paris*, Editions du CNRS & Cambridge, Mass.:MIT Press, 1995.

[11] P. Geach, What actually Exists?, in *Proc. of the Aristotelian Society*, Supplementary Volumes, Vol. 42, pp. 7-16, 1968.

[12] N. Guarino, Local Qualities, Quality Fields, and Quality Patterns: A Preliminary Investigation, in O. Kutz, M. Bhatt, S. Borgo & P. Santos (eds.), *Proc. of the Second Interdisciplinary Workshop SHAPES 2.0* (pp. 75-81), Rio de Janeiro, Brazil, 2013. Publié en ligne à : http://ceur-ws.org/Vol-1007/paper5.pdf

[13] G. Kassel, Processes Endure, Whereas Events Occur, in S. Borgo, R. Ferrario, C. Masolo & L. Vieu (eds.), *Ontology Makes Sense: Essays in honor of Nicola Guarino* (pp. 177-193), Frontiers in Artificial Intelligence and Applications, 136, IOS Press, 2019.

[14] G. Kassel, Physical processes, their life and their history, *Applied Ontology*, Vol. 15, N° 2, pp. 109-133, 2020.

[15] G. Kassel, Abstract events in semantics, *Philosophia*. Vol. 50, N° 2, pp. 1913-1930, 2022.

[16] G. Kassel, Plaidoyer pour des ontologies épistémiques, dans F. Saïs (ed.), *Actes des 33es Journées Francophones d'Ingénierie des Connaissances (IC) Plate-Forme Intelligence Artificielle (PFIA)* (pp. 48-55), 2022.

[17] A. de Libéra, Des accidents aux tropes. Pierre Abélard, *Revue de métaphysique et de morale*, Vol. 4, N° 36, pp. 479-500, 2002.

[18] E.J. Lowe, There Are (Probably) No Relations, in A. Marmodoro & D. Yates (eds.), *The Metaphysics of Relations* (pp. 100-112), Oxford University Press, 2015.

[19] F. MacBride, Relations, in E.N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy (Winter 2020 Edition)*, URL = https://plato.stanford.edu/archives/win2020/entries/relations/

[20] R. Mausfeld, The perception of material qualities and the internal semantics of the perceptual system, in L. Albertazzi, G. van Tonder & D. Vishwanath (eds.), *Perception beyond Inference. The Information Content of Visual Processes* (pp. 159-200), Cambridge, Mass: MIT Press, 2011.

[21] L. Naccache, *Le cinéma intérieur. Projection privée au cœur de la conscience*, Paris, Odile Jacob, 2020.

[22] F. Nef, Bergmann et l'ontologie de la connexion, in B. Langlet & J.-M. Monnoyer (eds.), *Gustav Bergmann. Phenomenological Realism and Dialectical Ontology* (pp. 157-172), De Gruyter, 2009.

[23] F. Nef, *L'Anti-Hume. De la logique des relations à la métaphysique des connexions*. Collection « Problèmes & Controverses », Paris, Vrin, 2017.

[24] F. Orilia & M. Paolini Paoletti, Properties, in E.N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2022 Edition), URL = https://plato.stanford.edu/archives/spr2022/entries/properties/

[25] E. Pacherie, Modes de structuration des contenus visuels, in J Bouveresse & J.-J. Rosat (eds.), *Philosophies de la perception* (pp. 263-289), Odile Jacob, 2003.

[26] G. Rodriguez-Pereyra, *Resemblance Nominalism. A solution to the problem of universals*, Oxford: Clarendon Press, 2002.

[27] B. Russell, *Principles of Mathematics*, Cambridge, UK: Cambridge University Press, 1903.

[28] S. Soames, Cognitive propositions, in J.C. King, S. Soames & J. Speaks (eds.), *New thinking about propositions* (pp. 91-124), Oxford: Oxford University Press, 2014.

[29] G.F. Stout, La nature des universaux et des propositions, dans E. Garcia & F. Nef (dir.), *Métaphysique contemporaine. Propriétés, mondes possibles et personnes* (pp. 121-142), Paris, Vrin, 2007 ; trad. par E. Garcia de The nature of Universals and Propositions, *Proceedings of the British Academy*, 10, pp. 157-172, 1921-3.

[30] A.L. Thomasson, Social Entities, in R. Le Poidevin, P. Simons, A. McGonigal & R.P. Cameron (eds.), *The Routledge Companion to Metaphysics* (pp. 545-554), London:Routledge, 2009.

[31] K. Twardowski, Sur la théorie du contenu et de l'objet des représentations, dans J. English (éd.), *Husserl – Twardowski, sur les objets intentionnels (1893-1901)*, Paris, Vrin, pp. 85-200, 1993 ; trad., introduction et notes par J. English de *Zur Lehre vom Inhalt und Gegenstand der Vorstellungen. Eine psychologogische Untersuchung*, Vienne, Hölder, 1894.

[32] R. VanRullen, A. Pascual-Leone & L. Battelli, The continuous wagon wheel illusion and the "when" pathway of the right parietal lobe: A repetitive transcranial magnetic stimulation study, *PLos One*, Vol. 3, N° 8, e2911, 2008.

[33] D.C. Williams, Les éléments de l'être, dans E. Garcia & F. Nef (dir.), *Métaphysique contemporaine. Propriétés, mondes possibles et personnes* (pp. 33-53), Paris, Vrin, 2007 ; trad. par F. Pascal & F. Nef de On the Elements of Being, *The Review of Metaphysics*, Vol. 7, N° 1, pp. 3-18, 1953.

Regard sur l'Ingénierie de la Connaissance face à l'ISO30401

Alain Berger

Ardans
6 rue Jean Pierre Timbaud, « Le Campus » Bâtiment B1,
78180 Montigny-le-Bretonneux, France
www.ardans.fr - aberger@ardans.fr

Résumé

Afin d'échanger entre chercheurs, théoriciens et praticiens de l'ingénierie de la connaissance, à la question de l'« IA et les normes », l'article propose une réflexion sur ce qu'est aujourd'hui l'ingénierie de la connaissance dans le monde « opérationnel » et comment cela s'inscrit par rapport à la norme ISO30401 *publiée en 2018. Cette vision est complétée par l'arrivée de l'environnement* PARNASSE *comme support à la mise en pratique de l'*ISO30401 *dédié au « manager de la connaissance ».*

Mots-clés

Ingénierie de la connaissance, Norme ISO30401 *, élicitation de la connaissance,* PARNASSE .

Abstract

In order to exchange between researchers, theorists and practitioners of knowledge engineering, to the question of « AI and standards », the article proposes a discussion on what is knowledge engineering in the « operational » world and how this relates to the ISO30401 *published in 2018. This vision is completed by the arrival of the* PARNASSE *environment as a support to the implementation of* ISO30401 *dedicated to the « knowledge manager ».*

Keywords

Knowledge engineering, ISO30401 *standard, knowledge elicitation,* PARNASSE .

1 De la connaissance dans l'IA jusqu'à la norme ISO30401

Il est toujours très intéressant de positionner l'« *Ingénierie de la Connaissance* » dans les différentes disciplines qui constituent l'intelligence artificielle. En partant de la conjecture de Dartmouth proposée en 1955, postulat selon lequel « *tous les aspects de l'apprentissage ou toute autre caractéristique de l'intelligence peuvent en principe être décrits si précisément qu'une machine peut être faite pour le simuler* »[13] la problématique de l'intelligence artificielle était posée comme « *une tentative ... faite pour trouver comment faire en sorte que machines utilisent le langage, forment des abstractions et des concepts, à résoudre des types de problèmes aujourd'hui réservés aux humains, et à s'améliorer d'elles-mêmes* ».

Près de soixante-dix ans plus tard, avec les progrès indéniables des techniques de l'apprentissage et des agents conversationnels, il est particulièrement intéressant d'observer les évolutions de cette discipline centrée sur les connaissances depuis les « *systèmes experts* » vers celle du « *Management de la Connaissance* » (ou *Knowledge Management*). Comment depuis la mise en œuvre de moteurs de règles, les réflexions sur leur implantation opérationnelle au sein d'organisation humaine, les apports de Ikujiro Nonaka[11] à Michel Grundstein[10] se retrouvent avec un certain niveau d'agrégation dans la norme ISO30401 [15] ?

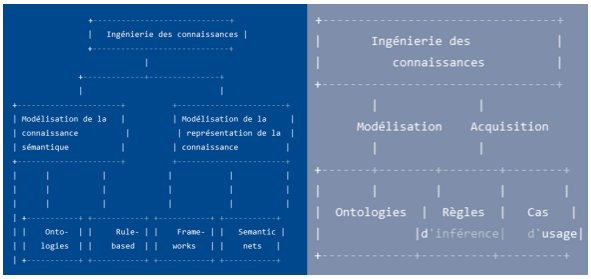


FIGURE 1 – Schémas représentant les techniques de modélisation en ingénierie des connaissances selon ChatGPT

2 L'ingénierie dédiée à la connaissance

La modélisation de la connaissance est un sujet qui ne cesse d'évoluer depuis les années quatre-vingts au fil des nouveaux formalismes et des langages proposés aux ingénieurs.

2.1 L'utilisateur contemporain

Pour le plaisir, deux parmi les schémas proposés par ChatGPT le 28 février 2023 (Cf. Figure 1) comme réponse à la question « *Aurais-tu un schéma qui représente les techniques de modélisation en ingénierie des connaissances ?* ». Au delà de la magie de réponses immédiates et intéressantes, l'usager est en droit de s'interroger lorsqu'en réitérant la requête *n* fois il obtient des réponses distinctes ; une interrogation inhérente à l'algorithme sous-jacent et génératrice du trouble. Le Pr Jean-Gabriel Ganascia [8] clarifie

radicalement le sujet et le qualifie d'ultracrépidarianiste [1]. Nous observons avec malice que le terme anglais utilisé pour solliciter ChatGPT est « *prompt* » ce qui se traduit selon l'humeur par « invite » ou « ordre ». Certains précisent qu'il convient de lui soumettre le « bon *prompt* » : l'utilisateur n'a qu'à bien se tenir !

2.2 Les formalismes disponibles

Entre les *frames*, les systèmes experts et leurs règles, la logique formelle et ses prédicats, les objets, les *blackboards*, les *truth maintenance systems*, les méta-connaissances, les réseaux bayésiens, les graphes de connaissances, les ontologies, les modèles de Markov, etc. à chacun son formalisme où deux objectifs étaient concourants voire concurrents : optimiser les performances et fournir la bonne réponse.

2.3 La question effective

Ainsi, si initialement il y avait une compétition sur les « *moteurs* », avec le temps celle-ci s'est déplacée vers la question de la qualité de la production et de l'accès à la connaissance. *In fine*, l'ingénieur de la connaissance se doit d'être humble pour retranscrire le plus fidèlement possible le savoir du sachant dans un système, et d'anticiper sur une question « libre » d'un utilisateur afin de lui procurer la meilleure réponse possible présente et qui lui est accessible dans le système (droit à en connaître) .

2.4 La production de la connaissance

Comment la connaissance se fabrique-t-elle ? Léonard de Vinci précisait que « *toutes nos connaissances ont pour origine notre perception* ». Oui, certes pour la génèse il y a une sorte d'intuition, mais la question de la preuve est essentielle comme le pressent Platon : « *la connaissance est une croyance vraie et justifiée* ». Nous n'évoquerons pas ici la question posée par Edmund Gettier qui s'interroge sur le fait que cela soit nécessaire et suffisant.
Il n'en reste pas moins que l'obsession de la justification, du calcul, de la démonstration, de la preuve anime sans relâche le scientifique. Le contexte et le processus qui conduisent à la connaissance sont essentiels ainsi que le rappelle Etienne Klein : « *Savoir, c'est en somme savoir 'comment on a su'* ». Ces éléments se révèlent comme fondateurs à la compréhension et surtout à la confiance dans le résultat délivré à l'utilisateur !

2.5 L'élicitation de la connaissance vivante

Comme ce que l'on appelle *connaissance* est aussi le fruit de l'expérience, elle n'a pas été forcément formalisée dans un cadre qui soit un dispositif de restitution de connaissance : elle est en tous les cas portée par l'humain au sein de son cerveau. La connaissance est bien incarnée. Le travail de l'ingénieur de la connaissance est de découvrir cette pépite et de la révéler, de l'extraire pour la mettre dans la forme la plus fidèle et conforme à ce que le sachant a pu exprimer lors des entretiens. En suscitant, en stimulant, en provoquant l'expert, l'ingénieur de la connaissance tire de l'expérience du sachant, lui fait sortir son savoir : il s'agit du processus d'élicitation de la connaissance. Cette démarche est très importante quand il s'agit de mettre à nu de la connaissance *implicite*, ce que soulève Jean-Yves Prax lorsqu'il mentionne le cas d'un expert surpris et s'exclamant : « *on ne sait pas ce qu'on sait !* » [14]. Pour être plébiscité dans

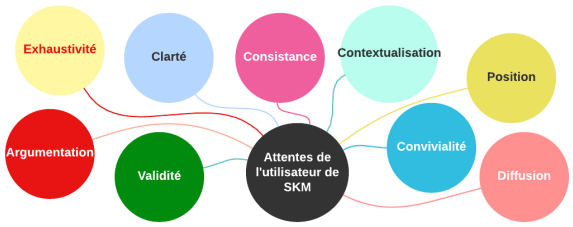


FIGURE 2 – Parmi les attentes d'un utilisateur de SKM

l'industrie, la base de connaissance ou le *Système de Management de la Connaissance* (SKM) doit satisfaire aux attentes des acteurs (Cf. figure 2) dont notamment :

- « *exhaustivité* » : il convient que la connaissance soit exhaustive sur le périmètre sur laquelle elle est porte afin d'obtenir la confiance de l'utilisateur à commencer par lui transmettre la réponse pertinente ;
- « *consistance* » : les résultats de « *navigation* » pour obtenir les contenus sont consistants ; cette stabilité rassure l'utilisateur ;
- « *clarté* » : les contenus sont clairs, dénués de toute ambiguïté, cela pour faciliter l'adhésion, l'appropriation et le bon usage par l'utilisateur,
- « *argumentation* » : les contenus sont argumentés et disposent des niveaux de preuve nécessaires pour une bonne appropriation par le lecteur ;
- « *contextualisation* » : il est fondamental de bien décrire le contexte dans lequel cette connaissance est valide pour être exploitée en toute sérénité ;
- « *position* » : l'élément de connaissance consulté est au cœur d'un réseau (implicitement sémantique) d'éléments de connaissance au sein desquels il doit être positionné dans une représentation cartographique multidimensionnelle : un réseau précieux pour évaluer la qualité de la base comme son homogénéité, ses relations, ses trous, ses densités ;
- « *diffusion* » : la connaissance est un actif précieux et est restreinte à ceux habilités à en connaître, celui qui en bénéficie doit savoir le mesurer ;
- « *convivialité* » : plus que jamais l'ergonomie d'un système à base de connaissance moderne doit être d'une ergonomie intuitive et fluide et démontrer qu'elle offre un retour sur investissement à l'usager sans pareil ;
- « *validité* » : la connaissance est vivante, comme elle s'affine dans le temps, elle est intrinsèquement « non monotone » et doit être datée ;

Pour l'organisme, le SKM a indubitablement la qualité de réaliser le *transfert de connaissance* vers l'utilisateur tel

1. « *Sutor, ne supra crepidam* », littéralement, le cordonnier (sutor), pas plus haut que la sandale (crepidam). Rapportée par Pline l'ancien dans son Histoire naturelle, cette sentence latine signifie que, de ce qui va au-delà de son métier, et que l'on ignore, on ne devrait parler.

que définit par l'équation de Davenport et Prusak[6] : *Transfer= Transmission + Absorption (and use).*

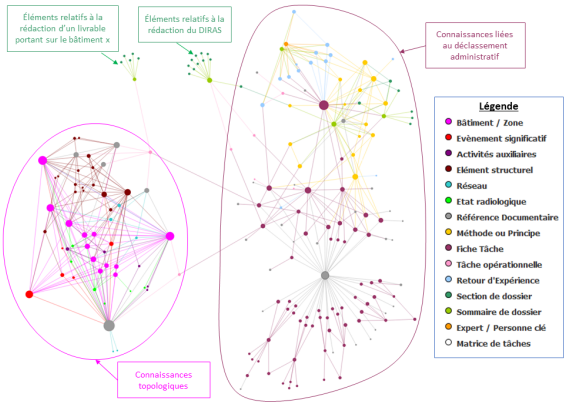


FIGURE 3 – La cartographie d'une base en construction

Ce qui est certain c'est qu'aujourd'hui les outils de modélisation de connaissance disposent d'environnement de visualisation (Cf. le graphe sous-jacent de la base[7] en figure 3) comme de fonctionnalités d'agrégation extrêmement riche pour distiller la bonne connaissance au bon moment à l'utilisateur (Cf. l'élément pointé par PARNASSE en figure 5).

2.6 La connaissance colligée implantée

Si comme l'exprime François Vexler[16]« *la connaissance, cela se mérite !* », en expert de l'« *Ingénierie de la Connaissance* » il sait que la clé d'un dispositif pérenne est de réaliser la bonne modularisation et structuration de la connaissance pour une exploitation vertueuse et fructueuse.
Il s'agit de trouver le juste équilibre entre une modularisation trop fine qui rebutera le futur contributeur pour alimenter et faire vivre dans le temps la base de connaissance, et une modularisation trop grossière qui ne guidera pas le futur lecteur pour trouver la réponse à sa question métier.
Le choix des modèles est d'autant plus aisé qu'il va coller à un processus métier, à une cinématique opérationnelle fussent-ils complexes.
La structuration, par ailleurs, s'impose en s'appuyant sur un langage métier partagé et une ontologie dénuée par construction de toute ambiguïté pour classer les concepts. Ces règles édictées[4, 12], la question de la conduite du changement et de la transmission du système à la maîtrise d'ouvrage devient prioritaire et c'est là que le risque se transfère vers la partie culturelle comme organisationnelle pour la dissémination de la démarche.

2.7 L'Ingénierie de la Connaissance

Actualiser la définition proposée en 2013[2] est nécessaire.
L'Ingénierie de la Connaissance est une discipline de l'IA qui couvre tout un cycle depuis l'*élicitation* d'un élément de connaissance, sa *structuration*, son *mûrissement* en termes de contenu, sa *description* (via une définition claire, non ambiguë, la rédaction étant appuyée par des illustrations ou schématisations si nécessaire), son *applicabilité* (en termes de domaine d'usage, de droit à en connaître en termes de publication ou d'habilitation, de durée de vie ou de date de péremption), et bien sûr de *validation* (appréciation d'expert, justification, degré de preuve).
Lorsque l'on travaille sur la mémoire collective d'un domaine métier, il convient d'orchestrer les différents **modèles** qui représenteront les éléments de connaissances, de poser les **liens de 'sémantique'** entre les éléments en relation, et les **liens de 'classification'** de ces éléments par rapport à des **ontologies** descriptives des concepts métier partagées, intelligibles, distinctes, complémentaires et non contestables.
L'« *Ingénierie de la Connaissance* » offre à l'**utilisateur 'lecteur'** les moyens de trouver la connaissance auquel il aspire et à se l'approprier, à l'**utilisateur 'contributeur'** la capacité à actualiser le patrimoine auquel il a accès, à à l'**utilisateur 'gestionnaire'** pour le compte de l'organisme, la capacité à exploiter son capital connaissance selon les règles de dissémination ou de protection en conformité à son attente ou à la règlementation.

3 Le Management des connaissances et l'ISO30401

3.1 La publication de la norme ISO30401

En novembre 2018, la norme ISO30401 intitulée « *Systèmes de management des connaissances – Exigences* »[15] est publiée par le Comité technique Management des Ressources Humaines de l'Organisation Internationale de Normalisation (ISO). Sa finalité est « *d'aider les organismes à concevoir un système de management qui valorise et facilite la création de valeur grâce aux connaissances* » (sic).
Ce qui est remarquable c'est que cette norme s'attaque de manière exhaustive à toutes les facettes du « *Management de la Connaissance* » .

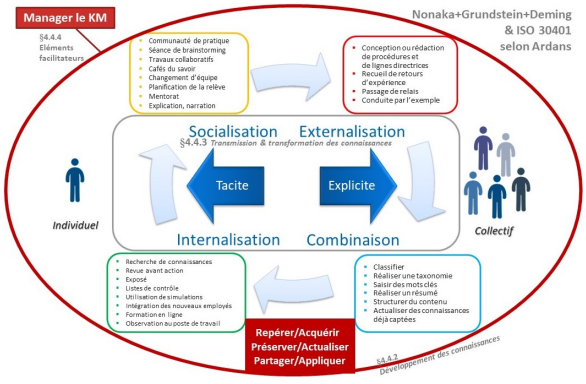


FIGURE 4 – Reformulation de l'ISO30401 selon Ardans

3.2 La première analyse de l'ISO30401

◇ **Le premier acquis** offert à tous par l'arrivée de cette norme est ce **vocabulaire commun** partagé par toute une communauté de professionnels de la discipline. Cette fondation est très importante quand on connait tout l'éventail

de provenance des contributeurs : culture, langue, métier, nature, taille et ressources de l'organisme.

◇ **Le deuxième avantage** est le fait que les trois grands fondamentaux classiques du « *Management de la Connaissance* » se retrouvent dans la norme (Cf. Le schéma de la reformulation par Ardans figure 4), à savoir :

1. la modélisation **SECI** de Nonaka & Takeuchi[11] (SECI pour Socialisation, Externalisation, Combinaison, Internalisation) avec le double positionnement Implicite/Explicite et Individuel/Collectif,
2. les cinq facettes de la **problématique de capitalisation des connaissances** de Grundstein[9] (Repérer, Préserver, Valoriser, Actualiser, Manager),
3. La roue de Deming **PDCA** (pour *Plan, Do, Check, Act* ou Planifier, Réaliser, Vérifier, Agir) relative à l'amélioration continue, et accompagnée par tout un arsenal d'éléments facilitateurs (humains, processus, technologies, gouvernance, culture).

◇ **Le troisième bénéfice** important est la vision de **S**ystème de **M**anagement de la **C**onnaissance (SMC ou SKM) avec toutes les dimensions de l'ingénierie système appliquée au KM comme l'indique Patrick Coustillière[5], pour être synthétique, ce qui est relatif aux **E**xigences, **R**essources, **O**rganisation [Rôle], **P**rocessus, **E**xports [Fournitures].

◇ **Le quatrième résultat** est astucieux car il délivre une clé pour les organismes en difficulté pour traiter le « *§7.1.6 Connaissances organisationnelles* » de la norme ISO9001:2015 relative au systèmes de management de la qualité, ce chapitre qui détient la palme de première non-conformité lors des audits. En précisant, qu'il s'agit d'une clé, nous entendons que si l'on trouve dans la norme les ingrédients, la recette n'y figure pas.

3.3 PARNASSE : vers une meilleure appropriation de la norme

3.3.1 L'expression de la finalité

Si dans certains pays des consultants se sont proclamés « auditeurs » de la norme ISO30401, en Europe la prudence est de rigueur. Pour autant, le Club Gestion des connaissances [2] a souhaité valoriser ses travaux internes autour de son « *SKM Book* » et le rendre plus accessible en tant que « *SKM de référence* » qui satisfait aux exigences de l'ISO30401. C'est ainsi que PARNASSE [3] a été conçu comme « *l'atelier du knowledge manager qui l'aide à évaluer et à améliorer leur système KM au regard de la norme ISO30401* » (Cf. Figure 5). L'objectif est de rendre possible l'exploitation « multi-vues » de ce référentiel complexe grâce à la puissance de modélisation de l'outil sélectionné : Ardans Knowledge Maker® [12]. La navigation dans la version logicielle PARNASSE [1] illustre clairement les vues du SKM Book. L'utilisateur sait ainsi :

- ⊙ Quelles activités du SKM Book afin de satisfaire aux exigences de la norme ?
- ⊙ Quelles exigences de la norme sont concernées par les activités du SKM Book ?
- ⊙ Pour une action KM, quelles activités du SKM Book sont préconisées et quelles exigences de la norme sont concernées ?

Avec PARNASSE, le Knowledge Manager sait évaluer les actions KM déjà en place, les situer au sein d'un SKM référence et ainsi identifier un plan de route vers une cible plus complète en accord avec la norme.

3.3.2 Une vision processus du SKM

L'axe de réflexion retenu par le Club Gestion des Connaissances pour aborder le SKM selon l'ISO30401 est indéniablement fondé sur les « Processus » ainsi que l'illustre le tableau ci-après issus de PARNASSE.

PARNASSE **: SKM de référence selon le Club GC**
Processus 1. Evaluer le contenu du patrimoine et le gérer
▷ P1.1 - Caractériser et évaluer le Patrimoine de connaissances
▷ P1.2 - Manager le Patrimoine de connaissances (qualité du contenu)
Processus 2. Faire vivre le patrimoine de connaissances et garantir son application
▷ P2.1 - Formaliser et mettre à disposition les connaissances
▷ P2.2 - Garantir l'application des connaissances
▷ P2.3 - Recenser les connaissances utiles à l'Organisation
▷ P2.4 - Gérer les Communautés de savoir et gérer l'expertise
Processus 3. Gérer et piloter les dispositifs d'acquisition de connaissances
▷ P3.1 - Processus RH - Recenser le besoin en formations nécessaires à l'activité (actuelle et future)
▷ P3.2 - Processus RH - Gérer et piloter l'apprentissage individuel (MOOC, e-learning, Coaching, ...)
▷ P3.3 - Gérer et piloter l'apprentissage en interaction collective (groupes d'expertises, séminaires, communautés d'apprentissage...)
▷ P3.4 - Définir les besoins en recrutement en lien avec les connaissances critiques de l'Organisation
▷ P3.5 - Processus RH - Gérer et piloter la construction des formations et solutions d'apprentissage
Processus 4. Soutenir les dispositifs de créativité et d'innovation
▷ P4.1 - Soutenir les activités de créativité
▷ P4.2 - Soutenir l'activité d'innovation
▷ P4.3 - Faire le bilan des connaissances acquises au cours des activités d'innovation / créativité
Processus 5. Soutenir les processus opérationnels
Processus 6. Transformer l'information externe en connaissance utile pour l'organisation
Processus 7. Outiller les activités KM
▷ P7.1 - Interagir avec les outils d'IA
Processus 8. Piloter le Système KM
▷ P8.1 - Définir la stratégie et les objectifs KM
▷ P8.2 - Construire le plan KM accepté par la direction de l'Organisation
▷ P8.3 - Évaluer le Système KM : les audits
▷ P8.4 - Superviser le Système KM : processus de décision, revues de pilotage, tableaux de bord des indicateurs, ressources humaines et matérielles, niveau de compétence...
▷ P8.5 - Organiser et conduire les actions de mise en place et d'amélioration du Système KM : sensibiliser, communiquer, mobiliser les acteurs, conduire les actions, ...

2. Association fondée en 2000 - https://www.clubgc-km.fr/

3. PARNASSE est l'acronyme de "**P**ortail **A**rticulant la **R**éférence **N**ormative iso30401 **A**vec un **S**ystème KM **S**tructuré pour l'**E**ntreprise" ou anglicisé en **P**ortal **A**rticulating the iso30401 **R**eference **N**orme via A **S**tructured KM **S**ystem for the **E**nterprise.

3.3.3 L'impact de l'ISO30401 sur la discipline

L'expert et concepteur de PARNASSE Daniel Colas considère qu'en donnant les clés et réponses à la compréhension du référentiel ISO30401 : « PARNASSE *démontre qu'une étape majeure de maturité dans la discipline est atteinte !* ».

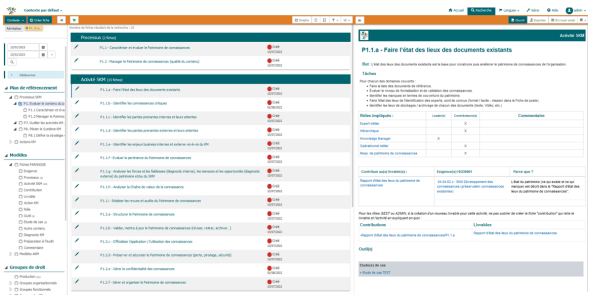


FIGURE 5 – PARNASSE aide à appréhender son SKM selon l'ISO30401 grâce à Ardans Knowledge Maker®

4 Perspectives

La question de l'utilisation des ressources disponibles sur le Web est un sujet en soi. La confiance dans la source et à la disponibilité d'éléments intègres dans le temps sont les obstacles majeurs. Dans le cadre industriel, la connaissance se consolide et se valide en interne afin d'en contrôler toutes les facettes qui garantissent et certifient la qualité et la confiance sur son contenu.

La question de la normalisation dans le domaine de l'intelligence artificielle est un sujet qui peut être traité par sous discipline ainsi que le démontre cette norme ISO30401 dédiée aux « *Systèmes de management des connaissances*».

Pour autant, il convient d'être prudent car si la norme a l'avantage de mettre à plat le langage commun des référents pour la discipline, elle ne donne aucune indication sur le comment faire, ni sur les pièges à éviter.

On note que des associations ou sociétés savantes référentes dans un domaine de l'IA sont aussi parfaitement capables de faire émerger des outils d'accompagnement à la compréhension fine de la norme. L'exemplarité de PARNASSE co-produit par le Club Gestion des Connaissances et Ardans en est la preuve pour le domaine de l'« *Ingénierie de la Connaissance* » et le « *Management de la Connaissance* » .

Comme le disait Alfred Korybski « *la carte n'est pas le territoire !* » ; si la norme peut couvrir l'ensemble des activités qui couvrent l'« *Ingénierie de la Connaissance* » , ce métier est extrêmement humain dans l'activité d'élicitation de la connaissance. En conséquence, des résultats toujours différents seront produits en fonction des professionnels qui l'exerceront.

La maîtrise technique de cette discipline reste aujourd'hui un art, dans la relation humaine, comme dans la transcription vers le système pour garantir la meilleure exploitation future en toute confiance.

5 Remerciements

Un article est aussi une histoire humaine et il me semble essentiel de citer et de remercier chaleureusement ceux qui m'ont invité et m'ont nourri par leurs contributions à cette production.

- Ce texte est une évolution de l'article initialement rédigé dans le cadre du Bulletin de l'AfIA numéro 120 pour le Dossier dont la thématique est « IA & Normes » dirigé par Nathalie Nevejans[3].
- Il a été possible grâce aux travaux réalisés par les consultants de l'équipe d'Ardans depuis 1999, avec en particulier, l'outil Ardans Knowledge Maker® avec sa méthode associée et ses nombreuses références industrielles acquises depuis.
- Les références sur PARNASSE sont en devenir ! Le Club Gestion des Connaissances et Ardans travaillent ensemble pour promouvoir l'outil comme un support du Knowledge Manager pour apprécier son SKM par rapport à l'ISO30401 : sont impliqués à la naissance cette jeune dynamique Daniel Colas, Patrick Coustillière, Aline Belloni, Jean-Pierre Cotton, Céline Fourtout, Olivier Rosnel et Grégory Elin.
- Enfin, une mention toute particulière à Jean Charlet. S'il ne m'avait pas convaincu, je n'aurai jamais osé proposer à la communauté scientifique d'IC2023 de discuter sur ces sujets sans contribution scientifique notoire. Les commentaires des relecteurs ont été des critiques extrêmement positives et constructives. Même si cette version actualisée intègre quelques précisions en réponse, je partage pleinement leur avis sur le fait que ce survol mérite des échanges en profondeur comme des travaux complémentaires sur de nombreux points.

Pour ceux qui en douteraient encore l'« *Ingénierie de la Connaissance* » est une discipline humaine extraordinaire et improbable où les rencontres comme les résultats le sont tout autant !

Références

[1] Aline Belloni and Daniel Colas. Parnasse : l'atelier du knowledge manager qui les aide à évaluer et à améliorer leur système km au regard de la norme iso30401. Digital Workplace Paris, 22 mars 2006.

[2] Alain Berger. L'ingénierie de la connaissance et la mémoire collective au cœur de la dynamique éthique des organisations. *Bulletin de AFIA n° 79*, pages 13–16, Janvier 2013.

[3] Alain Berger. L'ingénierie de la connaissance à l'heure de l'iso30401. *Bulletin de AFIA n° 120 - Direction Nathalie Nevejans - "IA & Normes"*, Avril 2023.

[4] Vincent Besson and Alain Berger. To initiate a corporate memory with a knowledge compendium : ten years of learning from experience with the ardans method. In *15èmes Journées Francophones Extraction et*

Gestion des Connaissances, EGC 2015, 27-30 Janvier 2015, Luxembourg, https://editions-rnti.fr/?inprocid=1002103, volume E-28, pages 401–412. Hermann-Éditions, 2015.

[5] Patrick Coustillière. L'ingénierie système, un outil pour le km manager ? In *Club Gestion des Connaissances*, volume https://www.clubgc-km.fr/articles/68927-05 of *Revues des Nouvelles Technologies de l'Information*, Mai 2022.

[6] Thomas Davenport and Laurence Prusak. *Working Knowledge : How Organizations Manage what They Know*. EBSCO eBook Collection. Harvard Business School Press, 1998.

[7] Céline Fourtout, Patrick Prieur, Alain Berger, Jean-Pierre Cotton, Aline Belloni, and Daniel Marx. Épione : Formaliser un processus métier par une démarche d'ingénierie de la connaissance : retour d'expérience sur le déclassement dans le nucléaire. In *9èmes Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle APIA2023, Strasbourg*, volume https://pfia23.icube.unistra.fr/conferences/apia, 6&7 Juil. 2023.

[8] Jean-Gabriel Ganascia. Le cordonnier, l'ultracrépidarianiste et chatgpt. In *Sciences et Avenir - Mai 2023*, volume https://lirelactu.fr/source/sciences-et-avenir/567d4840-b6d9-4e8b-9b78-fbc036cf8a4f, Mai 2023.

[9] Michel Grundstein. Développer un système à base de connaissance : un effort de coopération pour construire en commun un objet inconnu. In *Acte de la journée Innovation pour le travail en groupe*. CP2I, Novembre 1994.

[10] Michel Grundstein. "CORPUS," An Approach to Capitalizing Company Knowledge. In AIEM4 Proceedings., editor, *The Fourth International Workshop on Artificial Intelligence in Economics and Management*, Tel-Aviv, Israel, January 1996.

[11] Ikujiro and Hirotaka Takeuchi. *The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, 1995.

[12] Pierre Mariot, Christine Golbreich, Jean-Pierre Cotton, and Alain Berger. Méthode, Modèle et Outil Ardans de capitalisation des connaissances. In *RNTI E12 Modélisation des Connaissances*, volume https://editions-rnti.fr/render_pdf.php?p=1000709, pages 187–206, 2007.

[13] John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. A proposal for the dartmouth summer research project on artificial intelligence. In *http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf*, 1955.

[14] Jean-Yves Prax. *Manuel de Knowledge Management - 4èmeédition*. Les Actus du Savoir : Management/Leadership. Dunod, 2019.

[15] ISO Central Secretary. Knowledge management systems — requirements iso30401 :2018. In *https://www.iso.org/standard/68683.html*, International Organization for Standardization. Geneva, CH, 2018.

[16] François Vexler, Alain Berger, Jean-Pierre Cotton, and Aline Belloni. Eléments d'appréciation et d'analyse d'une base de connaissance : l'expérience industrielle d'Ardans. In *Actes Atelier AIDE à EGC'2013, 13ème Conférence Francophone sur l'Extraction et la Gestion des Connaissances*, volume https://eric.univ-lyon2.fr/aide/actesAIDE_EGC2013ENLIGNE.pdf, pages 59–72, Janvier 2013.

Améliorer la FAIRisation des données météorologiques à l'aide de la ressource lexicale INMEVO

Mouna Kamel[1,2], Nathalie Aussenac-Gilles[1], Cassia Trojahn[1,3]

[1] IRIT, CNRS, Université de Toulouse
[2] Université de Perpignan
[3] Université Toulouse 2 Jean Jaurès

prenom.nom@irit.fr

Résumé

Rendre les jeux de données météorologiques FAIR est un enjeu crucial pour la recherche scientifique. Le modèle ***dmo-core*** *permet, pour les données tabulaires, d'expliciter la sémantique des colonnes en les reliant à des concepts d'ontologies de domaine. Ces concepts étant généralement peu ou pas documentés, nous proposons (1) d'enrichir* ***dmo-core*** *afin de pouvoir associer aux colonnes leurs définitions issues d'une ressource lexicale, et (2) de générer une telle ressource, à partir du Vocabulaire Météorologique International de l'Organisation Mondiale de Météorologie (INMEVO).*

Mots-clés

Ressource lexicale, données FAIR, métadonnées sémantiques, données météorologiques.

Abstract

Making meteorological data sets FAIR is a key issue for scientific research. The ***dmo-core*** *model allows, for meteorological tabular data, to make the semantics of columns explicit by linking them to concepts of domain ontologies. As these concepts are generally little or not documented, we propose (1) to enrich* ***dmo-core*** *in order to associate definitions from a lexical resource to columns, and (2) to produce such a resource from the International Meteorological Vocabulary produced by the World Meteorological Organization (INMEVO).*

Keywords

Lexical resource, FAIR dataset, Semantic Metadata, Meteorological data.

1 Introduction

Les données météorologiques sont essentielles pour plusieurs types d'applications, dans de nombreux domaines tels que la météorologie, la climatologie, les transports, l'agriculture, le tourisme ou la médecine. Leur production est le fruit de modèles mathématiques qui intègrent des mesures issues de différentes sources, notamment des stations météorologiques, des satellites ou encore des radars météorologiques. Bien que ces données aient été mises à disposition en tant que données ouvertes sur différents portails, tels que des portails gouvernementaux (e.g. MeteoFrance [1], worldweather [2]), ou des portails associatifs ou privés (e.g. infoclimat [3] ou meteociel [4]), sous licences ouvertes, leur exploitation est plutôt limitée. Une des raisons est qu'elles sont décrites et présentées avec des propriétés pertinentes pour les experts du domaine de la météorologie (producteurs de données), mais qui ne sont pas forcément comprises et réutilisables par d'autres communautés scientifiques. Un moyen de rendre ces données accessibles (non seulement au niveau physique mais également au niveau logique) à des utilisateurs non experts du domaine, est de garantir leur conformité aux principes FAIR (Findability/Faciles à trouver, Accessibility/Accessibilité, Interoperability/Interopérabilité, Reusability/Réutilisabilité) en suivant les 15 recommandations qui leur sont associées [15]. L'adhésion aux principes FAIR s'impose à tout producteur de données qui veut garantir la réutilisation de ses données. Ces recommandations insistent entre autres sur la représentation formelle et sémantique des méta-données à l'aide de vocabulaires ou d'ontologies standards [2].

Dans ce contexte, le modèle sémantique (i.e. ontologie) **dmo-core** présenté dans [12] permet de décrire à la fois les données et les schémas de jeux de données, notamment les jeux de données tabulaires qui représentent la grande majorité des données ouvertes et qui sont disponibles principalement dans les formats CSV ou JSON. Une des spécificités de **dmo-core** est de donner la possibilité aux producteurs de données de sémantiser les colonnes des jeux de données en les reliant à des concepts d'ontologies de domaine. Par exemple, si on considère les colonnes *t* et *pmer* du jeu de données tabulaire SYNOP de Météo-France (voir Figure 1), et qui, pour les producteurs de données et spécialistes du domaine, correspondent à 'température' et 'pression de

1. https://donneespubliques.meteofrance.fr/
2. https://worldweather.wmo.int/en/home.html
3. https://www.infoclimat.fr/
4. https:www.meteociel.fr

la mer' respectivement, **dmo-core** permet de relier les colonnes *t* et *pmer* aux concepts *Temperature* et *SeaLevelPressure* de l'ontologie de domaine SWEET [5] (Semantic Web for Earth and Environment Technology Ontology).

numer_sta	date	pmer	tend	cod_tend	dd	ff	t	td	...
7005	2,02E+13	103180	-80	8	120	1.800000	274.350000	272.750000	...
7015	2,02E+13	103320	0	5	80	4.700000	275.250000	275.150000	...
7020	2,02E+13	102870	-70	8	80	1.300000	280.550000	279.450000	...
7027	2,02E+13	103080	0	0	100	4.200000	275.750000	275.750000	...
7037	2,02E+13	103190	-30	8	130	2.200000	272.250000	272.250000	...
7072	2,02E+13	103320	-20	8	60	1.100000	270.650000	269.550000	...
7110	2,02E+13	102740	10	0	180	0.600000	282.750000	282.650000	...
7117	2,02E+13	102760	-20	8	130	0.500000	281.550000	280.950000	...
7130	2,02E+13	102940	-90	8	110	3.100000	278.350000	278.050000	...
...	...	...	...	...	...	...	...	...	...

FIGURE 1 – Extrait du jeu de données SYNOP de Météo-France.

Force est de constater que les concepts des ontologies de domaine sont le plus souvent peu ou pas documentés. Par exemple, dans l'ontologie SWEET, la définition du concept *SeaLevelPressure* n'est donnée qu'en langue anglaise, alors qu'aucune définition de la notion de *temperature* n'est fournie. De plus, les termes synonymes des labels ne sont pas spécifiés. Or des outils comme FOOPS ! [5] intègrent dans l'évaluation du degré de FAIRisation des ressources sémantiques, notamment au niveau de la Réutilisabilité, les critères de documentation lisible par un humain et l'accès aux définitions des termes utilisés dans la ressource.

Dans l'optique d'améliorer la réutilisabilité (R de FAIR), l'idée est d'offrir la possibilité d'associer, lors de la description des colonnes, des ressources de type thesaurus, dictionnaire, lexique, etc. pour mieux documenter les colonnes des jeux de données tabulaires. Ceci suppose de disposer d'une ressource lexicale dans un domaine donné, et de pouvoir relier cette ressource au modèle d'annotation **dmo-core**. Pour le domaine de la météorologie, des ressources terminologiques existent comme la CF Standard Name Table développée par le groupe de travail Climate and Forecast (CF) Metadata Conventions [6], mais le lexique qui paraît le plus complet et surtout fait office de référence mondiale à notre connaissance, est le Vocabulaire Météorological International (WMO) produit par l'Organisation Météorologique Mondiale (OMM). Si l'on reprend l'exemple de la température, ce lexique fournit la définition suivante, et en quatre langues : *Grandeur physique caractérisant l'agitation moyenne des molécules dans un corps physique.* Bien que ce vocabulaire de l'OMM ait été intégré à la base de données terminologique UNTERM [7] créée par l'Organisation des Nations Unies, la ressource n'est malheureusement pas accessible pour pouvoir l'utiliser dans notre processus de FAIRisation. Le seul format disponible à notre connaissance est le format PDF.

Notre contribution pour améliorer le processus de FAIRisation des données météorologiques est donc double : (1) enrichir le modèle **dmo-core** pour pouvoir intégrer une ressource lexicale, et (2) créer une telle ressource à partir du vocabulaire de l'OMM disponible aujourd'hui au format PDF.

Cet article est organisé de la façon suivante. Après un état de l'art sur les vocabulaires proposés pour représenter les métadonnées (section 2), nous décrivons brièvement section 3 le modèle sémantique **dmo-core** sur lequel s'appuie notre proposition, ainsi que le processus de FAIRisation. Nous montrons section 4 comment enrichir le modèle **dmo-core** en offrant la possibilité d'intégrer une ressource lexicale. La section 5 est dédiée à la construction de la ressource lexicale météorologique INMEVO. Le processus de FAIRisation de données météorologiques à l'aide de **dmo-core** enrichi et du lexique INMEVO est décrit en section 6, et illustré par un exemple d'instanciation du jeu de données SYNOP de Météo-France. La dernière section dresse un bilan de ce travail et en présente les perspectives.

2 Etat de l'art

2.1 Représentation sémantique de schéma de métadonnées

De nombreux vocabulaires ont été proposés pour représenter les métadonnées de jeux de données, dont plusieurs sont devenus des standards de fait : Dublin core [8], VoID, Schema.org, DCAT [9], DCAT-AP. Il en existe des extensions pour pouvoir les adapter à des jeux de données spécifiques, comme GeoDCAT-AP pour les jeux de données géographiques ou StatDCAT-AP pour des données statistiques. D'autres approches utilisent des ontologies pour construire un schéma de métadonnées particulier. Ainsi, Parekh *et al.* [10] présentent un modèle de données et un mécanisme pour générer un schéma de métadonnées basé sur des ontologies. Au lieu de réutiliser les vocabulaires existants, les auteurs proposent leur représentation des métadonnées qui comporte des informations spatiales et temporelles, le contenu, la distribution et la présentation du jeu de données. D'une manière différente, Frosterus *et al.* ont étendu le vocabulaire VoID pour prendre en compte les jeux de données dans un format autre que RDF [4]. Au delà de ces initiatives qui ciblent tout type de jeux de données, plusieurs travaux proposent aussi des vocabulaires spécifiques à des domaines. Dans le domaine des Sciences Sociales et des Humanités du European Open Science Cloud (EOSC), c'est la Data Documentation Initiative [10] (DDI) qui est identifiée comme le principal standard. Cette initiative propose deux schémas XML, DDI-C et DDI-L, particulièrement adaptés aux données d'enquêtes quantitatives en sciences so-

5. https://bioportal.bioontology.org/ontologies/SWEET
6. https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html
7. https://unterm.un.org/unterm2/fr/
8. https://www.dublincore.org/specifications/dublin-core/dcmi-terms/
9. https://www.w3.org/TR/vocab-dcat/
10. https://ddialliance.org/learn/what-is-ddi

ciales et comportementales. DDI-L étend DDI-C pour décrire les jeux de données tout au long de leur cycle de vie. Plus récemment, la norme DDI-DCI (DDI Cross-Domain Integration) est adaptée à l'intégration de données provenant de différents domaines de recherche. DDI réutilise des vocabulaires tels que Prov-O, DC-terms, Data Cube ou CSVW et a des liens explicites avec des normes telles que DCAT. Cependant, il n'existe pas de sérialisation OWL de ces modèles.

Dans le domaine de la météorologie, les données étant de format tabulaire, plusieurs propositions ont utilisé RDF Data Cube [11] (qb) combiné à d'autres vocabulaires pour représenter les données d'observation. Dans [7], les auteurs combinent qb et l'ontologie de données de capteur SOSA [12] pour représenter 100 années de données relatives aux températures en RDF. Plus récemment, des données SYNOP ont été représentées en RDF à l'aide d'un modèle sémantique qui réutilise un ensemble d'ontologies existantes (SOSA/SSN, Time, QUDT, GeoSPARQL, et qb) [16].

Notre proposition consiste à enrichir le modèle sémantique **dmo-core** [12] capable déjà de représenter différents types de métadonnées au format tabulaire, dont le schéma de données et la structure du jeu de données.

2.2 Représentation de lexiques sous forme de graphes RDF

La représentation formelle de lexiques sous forme de données liées est considérée comme un enjeu pour leur diffusion, mais aussi pour améliorer l'interopérabilité d'autres ressources qui peuvent ainsi être enrichies de termes et de définitions [3]. Les recherches pour la représentation des lexiques sous forme de graphes RDF ont porté essentiellement sur trois aspects : (i) la définition de langages de représentation des lexiques permettant de les structurer et de les associer à des ontologies ; (ii) la construction de nouvelles ressources sous forme de graphes de connaissances ; ou encore (iii) la conversion de ressources existantes en RDF ou OWL [13]. Parmi les langages de représentation de lexiques, on trouve des recommandations du W3C comme SKOS [14] (Simple Knowledge Organization System) qui permet d'associer différents types de labels à des concepts ; des représentations en OWL de standards utilisés pour des lexiques, comme la représentation OWL du standard lexical SIMPLE [11] ; ou encore des propositions plus riches comme LEMON [8], fruit de plusieurs projets sur la représentation de ressources lexicales décrites par des connaissances linguistiques et liées à des ontologies. Parmi les ressources lexicales construites directement sous forme de graphes de connaissances, on peut citer BabelNet [15], une ressource multilingue basée sur une représentation riche des entrées lexicales, qui intègre WordNet et des éléments extraits de sources telles que Wikipedia [9]. Les premières ressources converties en langages du web sémantique sont WordNet et ses variantes locales (EuroWordnet etc.) traduits en OWL dès 2006 [14]. Depuis sa publication, le vocabulaire LEMON devient une norme pour une représentation riche de lexique. Ainsi, le lexique italien PAROLE SIMPLE CLIPS a été représenté à l'aide de LEMON [6] et publié sous forme de données liées, tout comme plus récemment, le dictionnaire Trésor de la Langue Française (TLF) [1].

Toutefois, peu d'articles traitent du processus de traduction ou de conversion du lexique vers un langage du web sémantique sous l'angle de la sémantisation d'un document pdf, comme c'est le cas dans cet article.

3 Travaux préalables

Nous rappelons brièvement ici les parties du modèle de **dmo-core** sur lesquelles repose notre proposition.

3.1 Le modèle dmo-core

Le modèle sémantique **dmo-core** a été élaboré dans le cadre du projet ANR Semantics4FAIR [16]. La Table 1 liste les espaces de nom et les prefixes associés utilisés dans ce modèle.

Table 1 – Espaces de nom des vocabulaires utilisés.

préfixe	espace de nom
dcat	http://www.w3.org/ns/dcat#
qb	http://purl.org/linked-data/cube#
skos	http://www.w3.org/2004/02/skos/core#
csvw	http://www.w3.org/ns/csvw#
prov	http://www.w3.org/ns/prov#
dmo-c	https://w3id.org/dmo#

Le modèle **dmo-core** [12] décrit, en OWL2, un schéma de métadonnées permettant la FAIRisation des jeux de données tabulaires, quel que soit le domaine. Il permet de décrire de manière fine le schéma des données et les structures de leurs distributions, et non de transformer les données en triplets RDF. Ce modèle est basé sur les vocabulaires DCAT [17] (`dcat`), CSVW [18]) et RDF Data Cube [19]), vocabulaires recommandés ou standardisés par le W3C. Tout jeu de données tabulaire, issu d'un catalogue (`dcat:Catalog`) et correspondant à une distribution (`dcat:Distribution`), peut être décrit selon ses différentes propriétés (`qb:MeasureProperty`, `qb:DimensionProperty` ou `qb:AttributProperty`), chacune correspondant à une colonne (`csvw:Column`) d'une table. Une colonne est reliée à une propriété par la relation `dmo-c:references`, et chaque propriété est reliée à un concept SKOS (`skos:Concept`) par la relation `qb:concept`. Ainsi, la sémantique de la colonne est fournie par ce concept qui peut lui-même correspondre ou être typé par un concept d'une ontologie du

11. https://www.w3.org/TR/eo-qb/
12. https://www.w3.org/2015/spatial/wiki/SOSA_Ontology
13. https://www.w3.org/TR/owl-guide/
14. https://www.w3.org/TR/skos-reference/
15. https://babelnet.org/
16. https://www.irit.fr/semantics4fair/index.html
17. https://www.w3.org/TR/vocab-dcat-2/
18. https://www.w3.org/ns/csvw/ (csvw
19. https://www.w3.org/TR/eo-qb/ (qb

domaine. Sur la Figure 7, le modèle **dmo-core** correspond aux concepts et relations en noir, bleu et orange.

3.2 dmo-core utilisé dans le domaine de la météorologie

L'utilisation de **dmo-core** pour un domaine particulier consiste à importer des ontologies de ce domaine. Dans le cas de la météorologie, nous avons importé les ontologies SWEET [20], ENVO [21] et SOSA [22]. Sur la Figure 7, ces ontologies de domaine sont représentées en violet. Le lien entre une colonne du jeu de données tabulaire et un concept d'ontologie de domaine devient effectif lors du processus de FAIRisation rappelé ci-dessous.

3.3 Processus de FAIRisation

Le processus de FAIRisation d'un jeu de données consiste à instancier **dmo-core** après importation des ontologies de domaine, par des propriétés propres à ce jeu de données. Il s'agit alors, pour chaque colonne Col_i de créer les instances suivantes :

1. l'instance INST_Col_i de `csvw:Column` ;
2. l'instance INST_i, instance à la fois de `skos:Concept` et du concept de l'ontologie de domaine ;
3. l'instance $\text{INST\_CompProp}_\text{i}$ (i.e., une dimension, un attribut ou une mesure) de `qb:ComponentProperty`.

Les instances $\text{INST\_CompProp}_\text{i}$ et INST_i sont reliées par la propriété `qb:concept`, et les instances $\text{INST\_CompProp}_\text{i}$ et INST_Col_i par la propriété `dmo-c:references`.

Afin d'illustrer le résultat de ce processus, nous donnons ci-dessous les métadonnées générées pour la colonne `pmer` du jeu de données SYNOP de Météo-France.

```
:sea_level_pressure_col
  rdf:type owl:NamedIndividual, csvw:Column ;
  dmo-c:references  :pmer_measure ;
  csvw:datatype "xsd:int" ;
  csvw:name "pmer" ;
  csvw:title "pression au niveau mer" .

:sea_level_pressure rdf:type
  owl:NamedIndividual ,
  sweet:SeaLevelPressure ,
  skos:Concept ,
  <http://www.w3.org/ns/sosa/ObservableProperty> .

:pmer_measure rdf:type owl:NamedIndividual ,
                       qb:MeasureProperty ;
  qb:concept :sea_level_pressure ;
  :unit_of_measures_attribute
        <http://qudt.org/vocab/unit#Pascal> .
```

4 Enrichissement de dmo-core

Bien que la sémantique des colonnes des jeux de données tabulaires soit explicitée par les concepts des ontologies de domaine, nombreuses sont les ontologies peu documentées, i.e. dont les concepts n'ont pas de définition en langage naturel. Dans ce cas, l'ontologie n'assure pas une bonne compréhension des données. Nous proposons donc d'enrichir **dmo-core** en offrant la possibilité de relier les colonnes des jeux de données tabulaires aux concepts définis dans une ressource sémantique de type thesaurus, dictionnaire ou lexique, qui complète ou apporte des définitions aux colonnes. Les vocabulaires SKOS [23] et OWL [24] (Web Ontology Language) permettent de publier et de rendre accessible tout vocabulaire. De plus, étant respectivement une recommandation et un standard du W3C, toute ressource formalisée à l'aide de ces vocabulaires adhère mieux aux principes FAIR.
Selon le vocabulaire SKOS, une instance de `skos:ConceptScheme` permet de représenter une ressource lexicale, une instance de `skos:Concept` une entrée lexicale, et la relation d'appartenance d'une entrée lexicale à la ressource par la relation `skos:inScheme`. Nous proposons alors d'intégrer ce modèle SKOS dans **dmo-core** en reliant le concept `dmoc:Dataset` (sous-classe du concept `dcat:Dataset`) au concept `skos:ConceptScheme` par la relation `dmo-c:isDocumentedBy`, pour indiquer le lien entre un jeu de données et la ressource lexicale qui peut être utilisée pour documenter les colonnes de ce jeu de données. Le modèle enrichi correspond à l'ensemble de la Figure 7, et l'intégration du lexique correspond aux relations et concepts en vert.

5 INMEVO : une ressource sémantique météorologique

Plusieurs étapes ont été nécessaires à la construction de la ressource lexicale INMEVO : analyse du document produit par l'OMM, identification du modèle des connaissances, puis extraction d'information et représentation des connaissances selon ce modèle.
Bien que le processus d'extraction des connaissances en français soit actuellement encore en cours de développement pour améliorer la qualité de la ressource, une version bêta de INMAVO est accessible à `https://gitlab.irit.fr/melodi/semantics4fair/inmevo`.

5.1 Analyse du document

Le *Vocabulaire météorologique international* [25] est une ressource terminologique publiée par l'*Organisation Météorologique Mondiale* (OMM) en 1966 dans une première version, puis en 1992 dans une version mise à jour et complétée. L'objectif était de normaliser la terminologie et de faciliter la communication entre experts de langues différentes. La version de 1992, la plus récente à ce jour, décrit environ 3500 termes météorologiques avec leurs définitions, dans 4 langues : anglais (EN), français (FR), russe (RU) et espagnol

20. `https://bioportal.bioontology.org/ontologies/SWEET`
21. `https://www.ebi.ac.uk/ols/ontologies/envo`
22. `https://www.w3.org/2015/spatial/wiki/SOSA_Ontology`
23. `https://www.w3.org/TR/swbp-skos-core-spec/`
24. `https://www.w3.org/TR/owl-guide/`
25. `https://public.wmo.int/fr/ressources/meteoterm`

(ES). Ces termes ont été approuvés par différentes organisations comme les membres de l'OMM, l'Aviation Civile Internationale ou la Commission Internationale de l'Eclairage. Bien que ce manuel manque de termes actuels, par exemple relatifs à la télédétection ou aux changements climatiques, il constitue un solide socle de connaissances pour comprendre les termes météorologiques ou interpréter les jeux de données du domaine.

Structure du document. Le document PDF est composé de 802 pages et comporte trois parties : les 19 premières pages sont consacrées à la préface, notice explicative, etc., les 694 pages suivantes sont propres à la terminologie, et les 89 dernières pages sont consacrées aux index (pour le français, l'espagnol et le russe). Chaque page est composée de 2 colonnes, une colonne par langue (pages anglais/français et pages russe/espagnol se succédant alternativement), la page pouvant contenir plusieurs entrées.

Dans ce lexique, chaque terme du vocabulaire météorologique possède une entrée lexicale, les entrées étant triées par ordre alphabétique des termes. Ces entrées lexicales bénéficient de propriétés lexicales, et de propriétés typographiques et dispositionnelles.

Propriétés lexicales. Deux types d'entrées sont à distinguer :

- Entrée de type 1 : c'est une entrée lexicale composée d'un identifiant, du terme défini que l'on appellera *terme descripteur* exprimé dans les 4 langues, d'une définition ou d'une liste de définitions exprimées dans les quatre langues, et pour chaque langue, d'une liste de synonymes en cours d'usage ou désuets s'il en existe. Le nombre de synonymes varie d'une langue à l'autre.
- Entrée de type 2 : c'est une entrée lexicale composée d'un identifiant, du terme défini que l'on appellera *terme descripteur* qui n'est exprimé qu'en anglais, et de la liste entre parenthèses des identifiants des entrées de type 1 ou 2 dont les termes descripteurs sont synonymes.

Dans l'exemple de la Figure 2, les entrées C0780 et C0790 sont de type 1, l'entrée C0800 est de type 2. L'entrée de type 2 C0800 (*clearance*) référence l'entrée C0820 de type 1 (Figure 3) qui possède le terme *clearance* parmi ses synonymes.

Par ailleurs, les entrées de type 2 ne concernent que les termes anglais. Pour les trois autres langues, la référence aux synonymes se fait via l'index qui liste les synonymes associés aux entrées terminologiques (Figure 4).

Il est important de noter que les entrées de type 2, dont le seul objectif est de permettre aux utilisateurs d'accéder à l'index alphabétique de tous les termes du lexique, y compris les synonymes, n'apportent pas de sémantique supplémentaire à celle exprimée au niveau des entrées de type 1. En effet, les termes descripteurs des entrées de type 2 existent déjà en tant que synonymes de termes descripteurs d'entrées de type 1. Par exemple, le terme descripteur *clearance* issu de l'entrée C0800 de type 2, référence l'entrée C0820 de type 1 dont le terme descripteur est *clearing*, qui a pour synonyme *clearance*, *clearing* et *clearance* partageant les mêmes définitions en tant que synonymes.

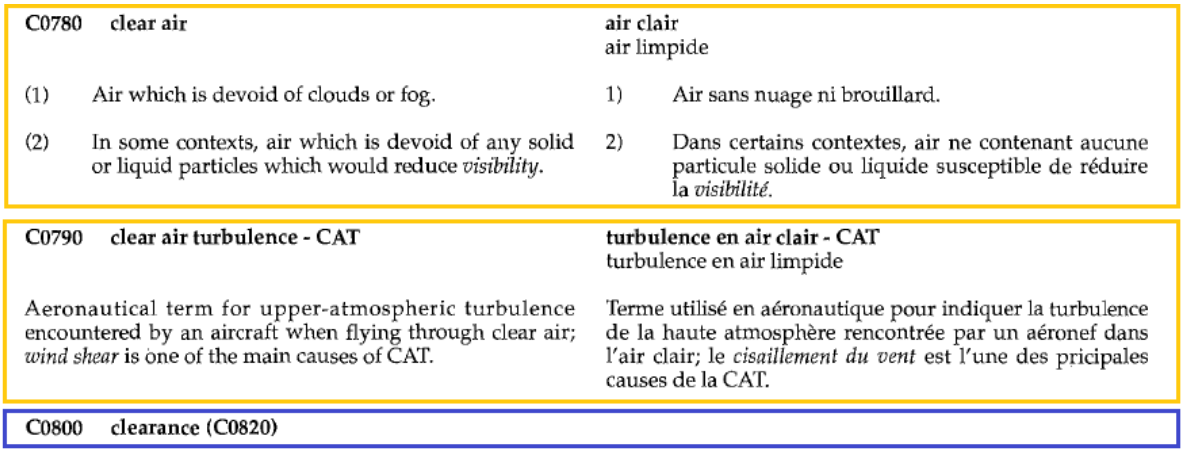
Propriétés typographiques et dispositionnelles. Des règles de mise en forme typographique et dispositionnelle appliquées sur la partie du document qui concerne la description des entrées lexicales permettent d'identifier les différents éléments (voir Figure 2) et d'envisager l'extraction automatique des connaissances exprimées dans cette ressource. Du point de vue typographique, les identifiants des entrées lexicales sont des chaînes de caractères gras répondant à un motif précis (une lettre majuscule suivie de 4 chiffres) ; les termes descripteurs sont en caractères minuscules et gras, les termes synonymes en caractères minuscules et séparés par des virgules, et entre crochets lorsqu'ils sont désuets ; les définitions commencent par une lettre majuscule et se terminent par un point, et sont numérotées lorsqu'un terme possède plusieurs définitions ; les termes apparaissant dans une définition et correspondant eux-mêmes à des entrées lexicales sont en italique ; les références aux concepts équivalents sont entre parenthèses.

Du point de vue dispositionnel, l'identifiant et le terme descripteur correspondant sont séparés par un caractère de tabulation ; les synonymes sont alignés verticalement aux termes descripteurs. À noter que la correspondance entre un identifiant et son terme descripteur en anglais est matérialisée par deux unités lexicales adjacentes dans le texte, alors que la correspondance avec les termes descripteurs dans les autres langues ne se fait que sur des critères dispositionnels.

Le processus d'extraction et de formalisation des connaissances que nous avons mis en œuvre est basé sur l'ensemble de ces règles. Dans cette étude, nous nous sommes limités à la représentation des connaissances exprimées en anglais et en français (présentes dans les pages paires) pour les raisons suivantes : la qualité des outils de conversion du format PDF au format texte est étroitement liée à la langue, et faute de connaissances en espagnol et en russe, nous n'aurions pas pu préjuger de la qualité des conversions. La méthodologie décrite ici pourra cependant être appliquée à la représentation des connaissances exprimées en russe et en espagnol une fois ces contraintes levées.

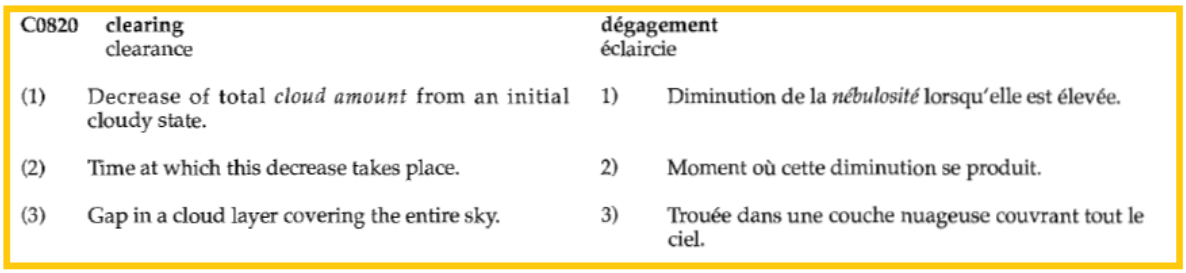
5.2 Représentation sémantique en SKOS : modèle IMV

Nous avons formalisé les entrées de type 1 du lexique de l'OMM et leurs propriétés à l'aide du modèle RDF et des vocabulaires SKOS et OWL. Chaque entrée lexicale de type 1 est représentée sous forme d'un concept SKOS, ayant pour URI l'identifiant de l'entrée lexicale. Ce concept SKOS est relié à plusieurs



C0780	**clear air**	**air clair** air limpide	
(1)	Air which is devoid of clouds or fog.	1)	Air sans nuage ni brouillard.
(2)	In some contexts, air which is devoid of any solid or liquid particles which would reduce *visibility*.	2)	Dans certains contextes, air ne contenant aucune particule solide ou liquide susceptible de réduire la *visibilité*.
C0790	**clear air turbulence - CAT**	**turbulence en air clair - CAT** turbulence en air limpide	
Aeronautical term for upper-atmospheric turbulence encountered by an aircraft when flying through clear air; *wind shear* is one of the main causes of CAT.		Terme utilisé en aéronautique pour indiquer la turbulence de la haute atmosphère rencontrée par un aéronef dans l'air clair; le *cisaillement du vent* est l'une des pricipales causes de la CAT.	
C0800	**clearance (C0820)**		

FIGURE 2 – Exemple d'entrées lexicales de types 1 et 2 (page 110 du document).



C0820	**clearing** clearance	**dégagement** éclaircie	
(1)	Decrease of total *cloud amount* from an initial cloudy state.	1)	Diminution de la *nébulosité* lorsqu'elle est élevée.
(2)	Time at which this decrease takes place.	2)	Moment où cette diminution se produit.
(3)	Gap in a cloud layer covering the entire sky.	3)	Trouée dans une couche nuageuse couvrant tout le ciel.

FIGURE 3 – Entrée lexicale C0820.

chaînes de caractères : le libellé de l'entrée (terme descripteur), ses éventuels synonymes actuels et désuets, par les relations `skos:prefLabel`, `skos:altLabel` et `skos:hiddenLabel` respectivement. On lui associe également ses définitions (multiples et dans les différentes langues) par la relation `skos:definition`. Le lien sémantique existant entre un concept et les concepts intervenant dans ses propres définitions (termes en italique qui correspondent à des entrées lexicales) est représenté par la relation `skos:related`.
Quant aux entrées de type 2, elles ne seront pas formalisées. Comme dit précédemment, ces entrées utiles pour constituer un index alphabétique, ne font que référencer des termes descripteurs déjà présents en tant que synonymes dans des entrées de type 1 (référencées entre parenthèses), qui possèdent les définitions. La Figure 5 présente le modèle IMV.
La section suivante mentionne les étapes nécessaires à l'extraction des connaissances du lexique de l'OMM au format PDF, et à la génération de la ressource lexicale INMEVO selon le modèle IMV.

5.3 Extraction et représentation des connaissances

La mise en forme typographique et dispositionnelle du document (notamment le parallélisme des paragraphes présents dans les colonnes) est primordiale pour caractériser les différents éléments du lexique à mettre en correspondance. Pour l'exploiter, différents outils de conversion de document PDF au format texte ont été testés, comme les bibliothèques python *Tesseract* et *pdfminer*, ou encore des logiciels en ligne comme *Adobe*, *OnlineOCR*, *PDFConvert*. Chacun de ces outils présente des inconvénients liés à la perte d'éléments typographiques et dispositionnels, comme les caractères accentués pour le français, le gras ou l'italique, ou encore en fusionnant les 2 colonnes en une seule (le plus souvent l'une au-dessous de l'autre). Nous avons finalement opté pour le convertisseur en ligne *PDFconverter* qui nous a semblé le plus fiable au regard du taux d'erreurs observé. Cet outil fournit un document au format Word dans lequel texte, typographie (caractères accentués, gras et italiques) et indices dispositionnels (tabulations, retours à la ligne, parallélisme des deux colonnes, etc.) sont globalement conservés. Néanmoins, des pertes de mise en forme ont été observées par endroit, et aucune bibliothèque exploitant des documents Word de notre connaissance n'ont permis d'exploiter les deux colonnes en parallèle.
De fait, deux pages Word, une correspondant à la colonne EN et l'autre à la colonne FR, ont été produites à partir de chaque page Word issue du convertisseur. Au final, nous avons obtenu 694 pages EN et 694 pages FR à exploiter, les ièmes pages FR et EN contenant les mêmes entrées lexicales. Cette réorganisation du corpus présente l'avantage de minimiser la propagation

français		russe		espagnol	
L0530	éclair	P0180	прошедшая погода	A0280	acdar
C0820	éclaircie	C0820	прояснение	C0820	aclaramiento
H0340	éclair de chaleur	P0940	прямая связь	A0220	aclimatación
S0940	éclair diffus	P0930	прямая связь одной точки с несколькими	C1210	acondicionamiento de aire

FIGURE 4 – Synonymes de l'entrée C0820 via les index.

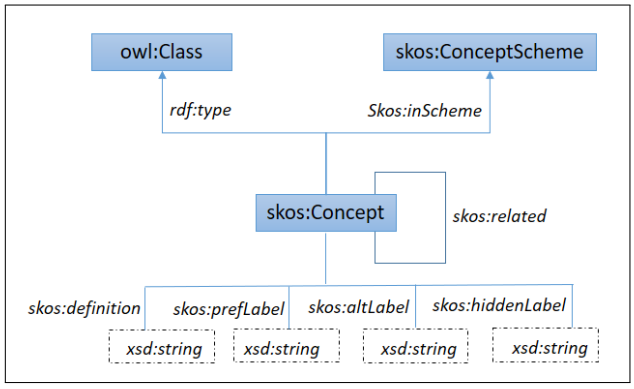


FIGURE 5 – IMV : Modèle sémantique pour la représentation du lexique de l'OMM.

des erreurs, lorsqu'il y en a, les erreurs se limitant à la seule page en cours de traitement.

Nous avons tout d'abord exploité les pages EN, les identifiants étant directement associés aux termes anglais. Nous donnons ci-dessous l'exemple de l'entrée lexicale C0780 du lexique de l'OMM, extraite et représentée selon le modèle IMV décrit Figure 5.

```
inmevo:C0780 rdf:type skos:Concept ;
  skos:prefLabel "clear air" @en ;
  skos:definition "Air which is devoid of clouds
    or fog." @en ;
  skos:definition "In some contexts, air which is
      devoid of any solid or liquid particles
      which would reduce oisibility." @en ;
  skos:related inmevo:V0390 .
```

Les pages en français ont été exploitées dans un deuxième temps. Les termes descripteurs n'étant plus associés à leurs identifiants dans les pages en FR, et l'OCRisation n'ayant pas permis de conserver un strict parallélisme (notamment en introduisant des sauts de lignes additionnels) entre les pages en EN et en FR, les principes d'extraction de connaissance mis en œuvre pour les pages en EN n'ont pas pu être appliqués.
Un moyen de retrouver la correspondance entre paragraphes EN et FR est de calculer une similarité sémantique entre eux. La méthode éprouvée dans cette version bêta de la ressource INMEVO repose sur une mesure de similarité calculée entre un paragraphe EN et la traduction en anglais du paragraphe FR. Pour cela, nous avons eu recours à l'API DeepL [26] pour la traduction, et à la distance de Levenshtein pour évaluer la similarité. Cette méthode a fourni de bons résultats du fait que la correspondance était recherchée entre les entrées lexicales figurant sur une seule page du corpus, et non pas sur la totalité du corpus.
Nous donnons ci-dessous l'exemple de l'entrée C0780 enrichie par les termes et définitions en français :

```
inmevo:C0780 rdf:type skos:Concept ;
  skos:prefLabel "clear air" @en ;
  skos:prefLabel "air clair" @fr ;
  skos:definition "Air which is devoid of clouds
    or fog." @en ;
  skos:definition "In some contexts, air which is
      devoid of any solid or liquid particles
      which would reduce oisibility." @en ;
  skos:definition "Air sans nuage ni
      brouillard." @fr ;
  skos:definition "Dans certains contextes, air ne
      contenant aucune particule solide ou liquide
      susceptible de réduire la visibilité." @fr ;
  skos:altLabel ""air limpide " @fr ;
  skos:related inmevo:V0390 .
```

Au final, la ressource INMEVO [27] est représentée par une instance `inmevo:INMEVO` de `skos:ConceptScheme`, par l'ensemble des instances `inmevo:A0010`, `inmevo:A0020`, ... de `skos:Concept` représentant les entrées lexicales, chacune d'elles étant reliée à `inmevo:INMEVO` par la relation `skos:inScheme`.

```
inmevo:INMEVO rdf:type skos:ConceptScheme .

inmevo:C0780 rdf:type skos:Concept ;
  skos:prefLabel "clear air" @en ;
  ...
  skos:related inmevo:V0390 ;
  skos:inScheme inmevo:INMEVO.

inmevo:C0790 rdf:type skos:Concept ;
  skos:prefLabel "clear air turbulence – CAT" @en ;
  ...
  skos:inScheme inmevo:INMEVO.

...
```

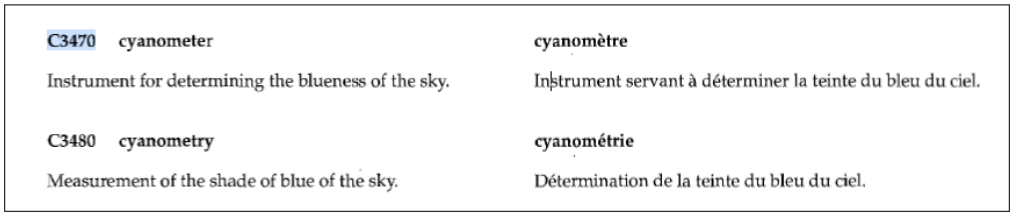
5.4 Evaluation de INMEVO

TABLE 2 – Evaluation de la ressource INMEVO. Les pourcentages indiquent les taux d'extractions correctes.

	Entrée de type 1
effectif	30
label@en	96%
label@fr	70%
def@en	96%
def@fr	86.6%
concepts reliés	84.2%
ocerisation @en	63%
ocerisation @fr	61%

26. https://www.deepl.com/

27. https://w3id.org/inmevo/



C3470	**cyanometer**	**cyanomètre**	
Instrument for determining the blueness of the sky.		Instrument servant à déterminer la teinte du bleu du ciel.	
C3480	**cyanometry**	**cyanométrie**	
Measurement of the shade of blue of the sky.		Détermination de la teinte du bleu du ciel.	

FIGURE 6 – Exemple d'entrées lexicales ayant conduit à des erreurs lors du processus d'extraction.

Nous avons sélectionné de façon aléatoire 30 entrées de type 1 de la ressource INMEVO, et les avons comparées manuellement aux entrées correspondantes du lexique de l'OMM. Cette comparaison a porté sur l'exactitude des labels extraits en français et en anglais, des définitions extraites en français et en anglais, des liens entre concepts (exprimés dans les définitions), ainsi que des erreurs d'OCRisation. Les résultats sont présentés Table 2.

Les erreurs générées lors du processus d'OCRisation sont pour la majorité dues à une mauvaise reconnaissance de caractères accentués pour le français, et aux caractères spéciaux utilisés dans les unités de mesure ou les formules mathématiques pour les deux langues. Par ailleurs, nous remarquons que les erreurs d'extraction sont plus fréquentes pour le français. Ce phénomène peut s'expliquer par deux facteurs interdépendants :

- la traduction de termes spécifiques au domaine n'est pas toujours appropriée. Par exemple, la traduction de *sonde de battage* (entrée lexicale R0970) selon le traducteur DeepL est *threshing probe* alors que le terme anglais associée à cette entrée est *ramsonde*.
- la correspondance entre un paragraphe anglais et celui issu de la traduction d'un paragraphe français est établie à l'aide d'une mesure de similarité entre chaînes de caractères, ce qui, lorsque la page comporte des entrées lexicales référençant des termes de la même famille, est source d'erreur.

L'exemple de la figure 6 illustre ces phénomènes, avec deux entrées lexicales C3470 et C3480 appartenant à la même famille (*cyanometer* et *cyanometry*), et une similarité plus élevée (selon la distance de Levenshtein) entre *Instrument for determining the shade of blue in the sky* (traduction obtenue pour *Instrument servant à déterminer la teinte du bleu du ciel*) et *Measurement of the shade of blue of the sky* qui est la définition de l'entrée lexicale C3480, qu'avec *Instrument for determining the blueness of the sky.*

6 FAIRisation de données météorologiques à l'aide de dmo-core enrichi

Le processus de FAIRisation d'un jeu de données météorologiques revient à instancier le modèle **dmo-core enrichi**, en suivant d'abord le processus de FAIRisation de **dmo-core** décrit en détail dans [13], et rappelé brièvement en section 3.3. Au terme de cette étape, toute instance de `qb:ComponentProperty` et reliée à une instance de `csvw:Column` peut également être reliée à l'instance de `skos:Concept` correspondant à l'entrée lexicale définissant la colonne si celle-ci existe dans la ressource INMEVO, par la relation `qb:concept`.

L'exemple représentant la colonne `pmer` à l'aide du modèle enrichi devient :

```
inmevo:INMEVO rdf:type skos:ConceptScheme .
inmevo:S0470 rdf:type skos:Concept ;
  skos:prefLabel "sea-level pressure" @en ;
  skos:prefLabel "scanneur" @fr ;
  skos:definition "Atmospheric pressure at mean
    sea level calculated from the observed station
      pressure." @en ;
  skos:definition "Pression atmosphérique au niveau
    moyen de la mer calculée d'après la pression
    mesurée à la station." @fr ;
    skos:inScheme inmevo:INMEVO .

:sea-level_pressure_col
  rdf:type owl:NamedIndividual, csvw:Column ;
  dmo-c:references  :pmer_measure ;
  csvw:datatype "xsd:int" ;
  csvw:name "pmer" ;
  csvw:title "pression au niveau mer" .

:sea_level_pressure rdf:type
  owl:NamedIndividual ,
  sweet:SeaLevelPressure ,
  skos:Concept ,
  <http://www.w3.org/ns/sosa/ObservableProperty> .

:pmer_measure rdf:type owl:NamedIndividual ,
                       qb:MeasureProperty ;
  qb:concept :sea_level_pressure ;
  qb:concept inmevo:S0470 ;
  :unit_of_measures_attr
     <http://qudt.org/vocab/unit#Pascal> .
```

7 Bilan et Perspectives

L'objectif d'améliorer le degré de FAIRisation (et plus précisément le degré de Réutilisabilité) des jeux de données météorologiques en documentant les colonnes des jeux de données tabulaires nous a conduit à enrichir le modèle **dmo-core** pour pouvoir intégrer une ressource lexicale exprimée en SKOS dans le schéma

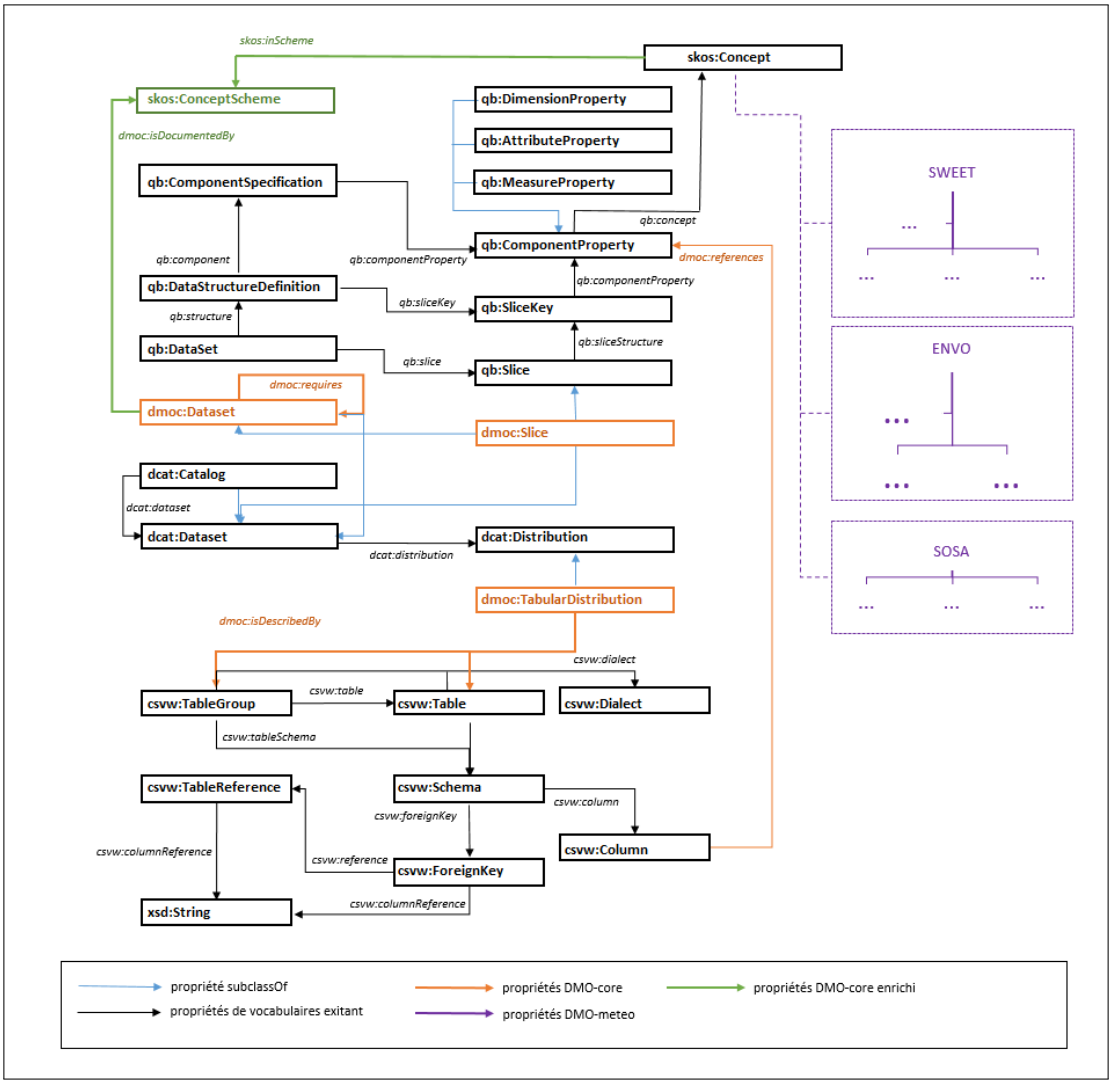


FIGURE 7 – Extension de dmo-meteo à l'aide d'une ressource lexicale au format SKOS.

d'annotation, et à construire la ressource lexicale INMEVO (actuellement dans une version bêta) spécifique au domaine de la météorologie. INMEVO est issue du Vocabulaire International de Météorologie produit par l'Organisation Mondiale de Météorologie, dont les entrées lexicales expriment des termes en quatre langues, des définitions riches et des termes synonymes. La ressource INMEVO est accessible et peut être utilisée dans diverses applications. Par ailleurs, l'approche proposée est générique, et peut être étendue à tout jeu de données, quel que soit le domaine, dès lors qu'une ressource lexicale du domaine formalisée en SKOS est disponible.

Plusieurs suites à cette étude sont envisagées. Un premier objectif est d'améliorer la qualité de la ressource INMEVO, (1) en améliorant le processus d'extraction notamment pour le français, afin de limiter les erreurs mentionnées dans la section 5.4, (2) en rajoutant des informations présentes dans le lexique et non encore prises en compte, comme le pays d'usage d'un terme (e.g. CA pour le Canada), et (3) en intégrant les vocabulaires espagnol et russe présents dans le lexique. Une vérification manuelle au final serait dans tous les cas nécessaire. Au delà du Vocabulaire International de Météorologie produit par l'OMM, cette ressource pourrait être mieux organisée en différenciant par exemple les phénomènes (e.g. dégagement, éclaircie) des propriétés de phénomènes (e.g. pression atmosphérique, niveau de la mer). Le deuxième objectif est de proposer des méthodes d'alignement automatique entre les concepts des ontologies de domaine (e.g. SWEET) et les concepts SKOS du schéma de concepts `inmevo:INMEVO`, toujours dans la perspective de documenter cette fois les ontologies du domaine de la météorologie.

Remerciements

Ce travail a bénéficié du soutien financier de l'ANR pour le projet Semantics4FAIR (2019-2022), contrat

ANR-19-DATA-0014-01.

Références

[1] S. Ahmadi, M. Constant, K. Fort, B. Guillaume, and J. P. McCrae. Convertir le trésor de la langue française en ontolex-lemon : un zeste de données liées. In *Journées LIFT 2021, Linguistique informatique, formelle et de terrain*, Grenoble, France, 2021.

[2] E. Amdouni and C. Jonquet. FAIR or FAIRer ? An integrated quantitative FAIRness assessment grid for semantic resources and ontologies. In *MTSR - 15th International Conference on Metadata and Semantics Research*. Springer, Nov. 2021.

[3] N. Calzolari. Approaches towards a "lexical web" : the role of interoperability. In J. Webster, N. Ide, and A. C. Fang, editors, *Proceedings of the First International Conference on Global Interoperability for Language Resources (ICGL 2008)*, pages 18—25. City University, 2008.

[4] M. Frosterus, E. Hyvönen, and J. Laitio. Datafinland - A semantic portal for open and linked datasets. In G. Antoniou, M. Grobelnik, and et al., editors, *8th Extended Semantic Web Conference, ESWC, Heraklion, Crete, Greece*, volume 6644 of *LNCS*, pages 243–254. Springer, 2011.

[5] D. Garijo, Ó. Corcho, and M. Poveda-Villalón. Foops ! : An ontology pitfall scanner for the FAIR principles. In O. Seneviratne, C. Pesquita, J. Sequeda, and L. Etcheverry, editors, *Proc. of the ISWC 2021 Posters, Demos and Industry Tracks : From Novel Ideas to Industrial Practice co-located with 20th Int. Semantic Web Conference (ISWC 2021)*, volume 2980 of *CEUR Workshop Proc.* CEUR-WS.org, 2021.

[6] R. D. Gratta, F. Frontini, F. Khan, and M. Monachini. Converting the parole simple clips lexicon into rdf with lemon. *Semantic Web*, 6 :387–392, 2015.

[7] L. Lefort, J. Bobruk, A. Haller, K. Taylor, and A. Woolf. A linked sensor data cube for a 100 year homogenised daily temperature dataset. In *Proc. of the 5th Int. Workshop on Semantic Sensor Networks*, volume 904, pages 1–16, 2012.

[8] J. McCrae, D. Spohr, and P. Cimiano. Linking lexical resources and ontologies on the semantic web with lemon. In *Proceedings of the 8th Extended Semantic Web Con- ference (ESWC 2011) on The Semantic Web : research and ap- plications - Volume Part I*, pages 245—259, Berlin, Heidelberg, 2011. Springer-Verlag.

[9] R. Navigli, M. Bevilacqua, S. Conia, D. Montagnini, and F. Cecconi. Ten years of babelnet : A survey. In *Proceedings of IJCAI 2021*, pages 4559–4567, 2021.

[10] V. Parekh, J. Gwo, and T. W. Finin. Ontology based semantic metadata for geoscience data. In H. R. Arabnia, editor, *Conference on Information and Knowledge Engineering*, pages 485–490, 2004.

[11] A. Toral and M. Monachini. Simple-owl : a generative lexicon ontology for nlp and the semantic web. In *Workshop of Cooperative Construction of Linguistic Knowl- edge Bases, 10th Congress of Italian Association for Artificial Intelligence (IA*AI), Rome (Italy)*, 2007.

[12] C. Trojahn, M. Kamel, A. Annane, N. Aussenac-Gilles, and B. L. Nguyen. A FAIR Core Semantic Metadata Model for FAIR Multidimensional Tabular Datasets. In O. Corcho, L. Hollink, O. Kutz, N. Troquard, and F. J. Ekaputra, editors, *23rd International Conference on Knowledge Engineering and Knowledge Management (EKAW 2022)*, volume 13514 of *Lecture Notes in Computer Science book series (LNCS)*, pages 174 – 181, Bolzano, Italy, Sept. 2022. Springer.

[13] C. Trojahn, M. Kamel, A. Annane, N. Aussenac-Gilles, B.-L. Nguyen, and C. Baehr. FAIRification of Multidimensional and Tabular Data by Instantiating a Core Semantic Model with Domain Knowledge : Case of Meteorology. In E. Garoufallou, M.-A. Ovalle-Perandones, and A. Vlachidis, editors, *16th International Conference on Metadata and Semantics Research (MTSR 2022)*, volume TBA, page à paraître, London, United Kingdom, Nov. 2022. springer.

[14] M. Van Assem, A. Gangemi, and G. Schreiber. Conversion of wordnet to a standard rdf/owl representa- tion. In *Proceedings of LREC2006*, Genova, 2006. ELRA, Paris.

[15] M. Wilkinson, M. Dumontier, and *et al.* Evaluating FAIR maturity through a scalable, automated, community-governed framework. *Sc. Data*, 6(1) :1–12, 2019.

[16] N. Yacoubi, C. Faron, F. Michel, F. Gandon, and O. Corby. A Model for Meteorological Knowledge Graphs : Application to Météo-France Observational Data. In *22nd Int. Conf. on Web Engineering, ICWE 2022*, Bari, Italy, July 2022.

Session 7 : Peuplement d'ontologies et annotation sémantique

Peuplement d'ontologie à partir de petites annonces immobilières

Céline Alec[1]

[1] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

celine.alec@unicaen.fr

Résumé

Le peuplement d'ontologie à partir de textes vise à transformer du contenu textuel en assertions ontologiques. Cela permet d'obtenir une représentation structurée du contenu d'un texte et d'automatiser sa compréhension. Cet article traite d'une approche de peuplement automatique d'une ontologie à partir de petites annonces immobilières. Celle-ci s'appuie à la fois sur une analyse textuelle et une analyse des connaissances du domaine. Des expérimentations, réalisées sur des annonces françaises de ventes de maisons, sont discutées et donnent des résultats encourageants.

Mots-clés

Peuplement d'ontologie, Traitements et raisonnement sur des connaissances, OWL

Abstract

Ontology population from texts transforms textual contents into ontological assertions. A text content has then a structured representation, enabling its understanding. This article deals with an approach to automatically populate an ontology from real estate classified ads. This approach is based on both a text-based and a domain knowledge-based analysis. Experiments, carried out on French house sale ads, are discussed and give encouraging results.

Keywords

Ontology population, Knowledge processing and reasoning, OWL

1 Introduction

Les ontologies [18] permettent de stocker et de partager les connaissances d'un domaine. Elles comportent une hiérarchie de concepts et des relations entre ces concepts. Le processus d'ajout d'instances à une ontologie est appelé peuplement d'ontologie. Une ontologie peuplée peut également être appelée base (ou graphe) de connaissances.

L'approche proposée dans cet article fait partie du projet DECA (Détection d'Erreurs et Correction d'Annotations). Ce projet s'intéresse à des descriptifs annotés, c'est-à-dire des descriptions textuelles, auxquelles sont ajoutées des annotations. Par exemple, c'est le cas des petites annonces, annotées avec les critères auxquels elles répondent. Les annotations sont théoriquement censées décrire les caractéristiques de l'objet ou de l'événement décrit dans la description. Cependant, ce n'est pas toujours le cas. En effet, on observe fréquemment des annotations erronées, soit à cause de fautes de frappe, soit à cause d'un « détournement d'usage » : les descriptions sont délibérément mal annotées afin d'augmenter leur visibilité. Par exemple, on peut trouver une annonce immobilière dont la ville annotée est X, alors que la description indique « à 30 minutes de X » ; ou dont l'annotation de la superficie est de 150 m^2 alors que la description indique « 147 m^2 ». Ces mauvaises annotations font perdre un temps considérable aux utilisateurs, qui doivent trier les multiples réponses répondant en théorie à leurs critères de recherche. Le projet DECA s'attaque à ce problème. Son objectif est de détecter et de corriger automatiquement les annotations erronées grâce aux incohérences qui peuvent être trouvées en confrontant les annotations et la description textuelle. Une telle confrontation nécessite d'assimiler les éléments essentiels du texte, autrement dit, de comprendre le texte comme le ferait un humain. C'est sur ce sous-problème que se concentre notre contribution. Notre objectif est de représenter le contenu des descriptions textuelles d'objets dans une ontologie de domaine. Il s'agit donc d'un problème de peuplement d'ontologie à partir de descriptions textuelles. Notre premier cas d'utilisation concerne des annonces de vente de maisons, mais l'approche proposée doit être aussi générique que possible, c'est-à-dire qu'elle doit être capable de peupler une ontologie de domaine à partir de descriptions dans de nombreux domaines, qu'il s'agisse de petites annonces (vente de véhicules, mode, etc.), ou de descriptions d'un certain type d'objet (restaurants, hôtels, etc.). Une ontologie du domaine concerné est considérée en entrée ; sa conception n'est pas l'objet de notre travail.

Le reste du document est organisé comme suit : la Section 2 présente les travaux connexes et positionne notre contribution. La Section 3 décrit notre approche. Les expérimentations sur notre cas d'utilisation sont détaillées dans la Section 4. La Section 5 conclut et suggère des travaux futurs.

2 Travaux proches et positionnement

Le peuplement d'ontologie a été étudié dans divers articles scientifiques. L'étude [12] présente un aperçu des travaux sur ce sujet. Dans cette section, nous nous concentrons uniquement sur les approches visant à extraire des informations de documents textuels non structurés et d'une ontologie de domaine donnée en entrée, en permettant l'ajout d'instances dans cette dernière, en particulier l'ajout d'assertions de propriétés. Les systèmes existants sont basés

sur diverses méthodes à base de règles utilisant des modèles lexico-syntaxiques (cf. paragraphe suivant) ; ou à base d'apprentissage automatique, [9, 19] ; ou encore sur des méthodes hybrides [3]. Certaines approches récentes utilisent l'apprentissage profond. Pour cela, les données textuelles sont exploitées pour produire un modèle de langage basé sur des plongements de mots. C'est le cas de [2] qui vise à peupler une ontologie dans le domaine biomoléculaire, ou de [6] qui utilise un LSTM pour peupler une ontologie traitant de cybersécurité. De façon générale, l'utilisation de techniques d'apprentissage automatique nécessite de disposer en amont d'une quantité suffisante de phrases et de leur correspondance ontologique, ce qui n'est pas le cas de nos données. Nous nous concentrons donc ici sur les approches qui exploitent des patrons lexico-syntaxiques.

L'approche ArtEquAKT [1] s'intéresse à du peuplement d'ontologie à partir du Web dans le domaine des artistes. Elle peuple l'ontologie avec des assertions de propriétés. Pour cela, le verbe trouvé dans une phrase entre deux instances de concept de l'ontologie est exploité. Sur le même principe, Makki [13] se concentre également sur les verbes afin de peupler l'ontologie avec des assertions de propriétés, mais l'approche est semi-automatique et indépendante du domaine. Une liste de verbes est extraite du corpus d'entrée pour chaque propriété de l'ontologie en utilisant Wordnet. Un ensemble de sept règles écrites manuellement est utilisé pour reconnaître les sujets et les objets d'une assertion de propriété potentielle. Les résultats sont ensuite validés par un expert. Dans [5], un framework est proposé pour instancier un concept et les relations qui le concernent. Tout d'abord, les entités nommées sont identifiées dans le texte (en exploitant également les co-références). Ensuite, des déclencheurs sont pris en compte, c'est-à-dire les noms de propriétés ainsi que leurs synonymes ; et des règles sont construites sur la base des phrases nominales précédées ou suivies d'un déclencheur. L'application de ces règles conduit au peuplement de l'ontologie. [16] présente une approche pour peupler une ontologie d'évènements criminels et leurs causes à partir de tweets de journaux espagnols. L'approche se base sur des patrons linguistiques. [11] a pour but d'extraire des assertions de propriété dans des documents règlementaires entre des incidents et des mesures (propriété « hasMeasure » et ses sous-propriétés) en se basant sur des règles. Les co-occurrences d'incidents et de mesures au sein d'une même phrase, paragraphe ou chapitre, sont utilisées ; mais cela ne permet pas de distinguer les sous-propriétés. Des patrons lexicaux sont employés dans ce cas. [15] présente une approche à base de règles pour extraire des relations à partir d'anecdotes musicales et peupler une ontologie. Elle exploite des règles basées sur des étiquetages grammaticaux. Le framework T2KG [10] se base sur des règles pour traduire du texte en triplets et utilise une approche hybride (règles et similarité) transformant les prédicats textuels en ceux d'un graphe de connaissances.

L'étude de ces approches montre plusieurs obstacles scientifiques émergents. En général, les méthodes à base de règles présentent une bonne précision au détriment du rappel [4]. Soit les règles sont propres à un domaine (cas des patrons lexicaux et de certains patrons syntaxiques très précis), soit elles sont génériques. Dans ce dernier cas, elles ne peuvent pas être aussi précises que des règles définies pour un seul domaine. Beaucoup de ces travaux nécessitent une intervention humaine pour valider les propositions trouvées. De plus, le verbe présente en général une grande importance dans le peuplement des propriétés, car il est fortement caractéristique d'une relation (par ex., « est marié à »). Enfin, la plupart des approches s'intéressent à des entités nommées en sujet et objet des propriétés, et se focalisent sur le peuplement des propriétés objet (object properties) au détriment des propriétés typées (data properties).

Dans notre contexte, nous souhaitons pouvoir appliquer une même approche sur plusieurs domaines, en restant néanmoins toujours dans le cadre des descriptions textuelles d'objets. Comme l'approche proposée sera la première étape en vue d'une correction automatique d'annotations, elle se doit d'être automatique, sans aucune validation humaine, et suffisamment générique. En général, les verbes ne sont pas très caractéristiques d'une relation dans les descriptions d'objets (par ex., « a » ou « possède »). Parfois, il peut n'y avoir aucun verbe (par ex., « 2 chambres, 1 salle de bain. »), ce qui complique le peuplement. Les propriétés objet et typées sont importantes. Les sujets et objets ne sont pas nécessairement des entités nommées. Enfin, notre contexte peut présenter des propriétés n-aires [1], qui représentent des notions complexes difficiles à peupler. Pour toutes ces raisons, les travaux cités ne sont pas adaptés à notre problématique originale, qui nécessite l'établissement d'une nouvelle approche.

3 L'approche KOnPoTe

Nous présentons KOnPoTe (Knowledge graph/ONtology POpulation from TExts), une approche pour peupler une ontologie de domaine à partir de descriptions textuelles d'éléments de ce domaine. Elle prend en entrée un corpus de descriptions ainsi qu'une ontologie du domaine et peuple l'ontologie en représentant les descriptions du corpus.

3.1 Les données initiales

L'ontologie initiale définit le domaine. Plus formellement, elle peut être définie comme un tuple $(\mathcal{C}, \mathcal{P}, \mathcal{I}, \mathcal{A}, \mathcal{R})$ où $\mathcal{C}$ est un ensemble de classes, $\mathcal{P}$ un ensemble de propriétés (objet et typées) caractérisant les classes, $\mathcal{I}$ un ensemble d'individus et d'assertions (potentiellement vide), $\mathcal{A}$ un ensemble d'axiomes représentables en OWL2 [2] et $\mathcal{R}$ un ensemble de règles SWRL [3] [8] (potentiellement vide). Elle peut être existante, construite manuellement ou (semi-) automatiquement. Sa conception ne fait pas partie de notre contribution. Le domaine y est représenté par une classe nommée ci-après *classe principale*. Les propriétés typées prises en compte peuvent utiliser des valeurs booléennes, numériques ou des chaînes de caractères. Elle peut contenir des individus initiaux, qui sont génériques. Chaque entité

1. Par exemple, exprimant qu'un bien se situe à une distance d'un lieu.
2. Web Ontology Language
3. Semantic Web Rule Language

de l'ontologie (classe, propriété, individu) possède un identifiant (URI) et éventuellement une terminologie plus avancée, via rdfs:label, rdfs:isDefinedBy, ainsi qu'une propriété d'annotation « unité », spécialement créée pour associer une entité à son unité ou à une expression d'unité. Cette particularité est exemplifié dans le paragraphe suivant. Le corpus utilisé en entrée est composé de documents qui décrivent chacun une instance de la *classe principale*.

FIGURE 2 – Vision partielle des entités de l'ontologie

L'approche est conçue pour être applicable à différents domaines de descriptions textuelles. L'exemple déroulé dans ce papier considère le domaine des ventes de maisons. La Figure 2 représente une vision partielle des classes et propriétés de l'ontologie utilisée dans ce cadre. Celle-ci contient des classes telles que la classe principale *Bien* (désignant un bien immobilier) ; des classes désignant des pièces comme *PièceDeMaison*, *Cuisine*, *Chambre* ; etc. Parmi les propriétés, on peut citer la propriété objet *seSituerA* reliant un bien immobilier à la commune dans laquelle il est situé, ou la propriété typée *surfaceEnM2* reliant une partie de bien (terrain, maison, etc.) ou une pièce à une valeur numérique. Les individus initiaux (non représentés sur la figure) sont génériques, par exemple, des instances de la classe *Commune*, ou de la classe *SystèmeDeChauffage*. L'ontologie contient également des axiomes, par exemple, le fait qu'un *Bien* ne peut être situé que dans maximum une *Commune*, ou le fait qu'une *Chambre* est disjointe d'une *Cuisine*. Enfin, certaines règles SWRL sont définies, par exemple, pour exprimer le fait que la propriété booléenne *séparé* doit être peuplée avec la valeur opposée de la propriété booléenne *ouvert*. La propriété *surfaceEnM2* est associée à son unité « m^2 », et la propriété *pourcentageHonoraires* à une expression d'unité « honoraires : xxx % ». Un changement dans la façon de représenter les unités (en utilisant, par exemple, une classe spéciale d'unités liée à un nom d'unité et à une valeur d'unité) peut être envisagé dans une version future.

3.2 Modélisation de l'approche

L'approche proposée se doit d'être applicable à différents domaines. Pour un domaine donné, il est nécessaire d'avoir en entrée une ontologie du domaine, ainsi qu'un corpus de documents, où chaque document est un texte qui décrit une instance de la *classe principale*. Ainsi, l'algorithme proposé ne doit pas dépendre de règles ou de modèles linguistiques basés sur le domaine. Nous avons donc choisi d'utiliser la terminologie du domaine (issue de l'ontologie), des indicateurs syntaxiques (comme les phrases ou l'ordre des expressions dans le texte) et des indicateurs de connaissances (comme les domaines et co-domaines de propriétés).
La Figure 1 montre les grandes lignes de l'approche. L'ontologie (O), ainsi que chaque document du corpus, sont utilisés (haut de la Figure) par un comparateur de terminologie. Cela conduit à des correspondances entre les mentions du texte et les entités de O (classes, propriétés et individus). Ensuite, un algorithme de peuplement est appliqué, pour obtenir l'ontologie peuplée. Cet algorithme de peuplement est une succession de plusieurs traitements (milieu de la Figure) : une initialisation d'un objet appelé « Traitements des correspondances » (TC) ; une instanciation de la *classe principale* ; une analyse textuelle, composée de divers traitements (bas de la Figure) ; ainsi qu'une analyse basée sur les connaissances de l'ontologie. Cela est appli-

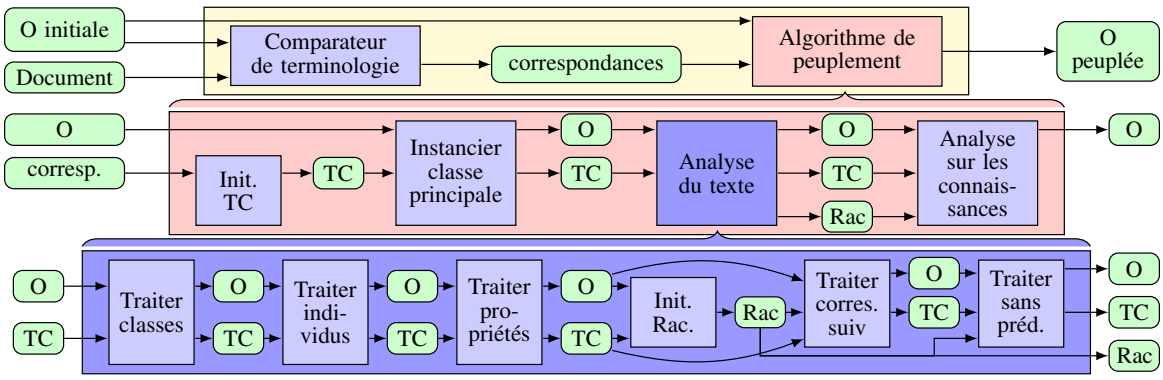


FIGURE 1 – Vision schématique de l'approche KOnPoTe

qué sur chaque document du corpus, permettant d'obtenir à la fin une ontologie représentant toutes les descriptions du corpus. La suite de cette section détaille chaque traitement en déroulant un exemple d'annonce de vente d'une maison.

3.3 Le comparateur de terminologie

Le comparateur de terminologie est la première étape de l'approche. Il prend en entrée l'ontologie initiale et un document, et produit des correspondances entre les mentions textuelles du document et les entités de l'ontologie (cf. Figure 1). Le texte est découpé en phrases et lemmatisé. Les mots-clés de l'ontologie (fragments d'URI, labels, unités) sont aussi lemmatisés. Des correspondances sont établies entre le texte et les mots-clés de l'ontologie. Celles-ci concernent également les expressions d'unité, par exemple, la mention « *honoraires : 4%* » correspond à l'expression d'unité « *honoraires :* xxx % ». Les inclusions sont ignorées. Par exemple, s'il existe une correspondance sur la mention « centre ville » et sur sa sous-mention « ville », seule celle sur « centre ville » est considérée.

Cormelles-le-Royal : Magnifique maison à 15 min du centre-ville de Caen et à 3 min à pied des commerces et des écoles, 110 m^2 sur un terrain de 400 m^2. Le rez-de-chaussée est composé d'une cuisine équipée, d'un salon avec cheminée, exposé sud-ouest, ainsi que d'une chambre de 15 m^2. Premier étage : 2 chambres et 1 sdb. Proche des transports en commun. Terrain arboré et clos. Honoraires : 4%.

FIGURE 3 – Exemple d'un texte et ses correspondances

La Figure 3 montre un exemple fictif d'une annonce immobilière. Les correspondances obtenues concernent des individus (ex : « rez-de-chaussée » ↔ *RDC*), des classes (« maison » ↔ *Maison*), des propriétés objet (« Proche des » ↔ *estProcheDe*) ou typées (« m^2 » ↔ *surfaceEnM2*).

3.4 L'algorithme de peuplement

Les correspondances sont fournies en entrée à l'algorithme de peuplement. Ce dernier comporte quatre tâches principales (cf. Figure 1), dont les deux premières sont du prétraitement. La troisième, appelée analyse du texte, est une tâche de peuplement utilisant les correspondances et des indicateurs textuels. La dernière tâche, appelée analyse sur les connaissances, vise à ajouter des assertions de propriétés en se basant sur les connaissances de l'ontologie. Ces tâches sont décrites dans la suite de cette section.

3.4.1 Les deux tâches de pré-traitement

Initialisation des traitements de correspondances La première tâche consiste à initialiser les traitements de correspondances $TC = \{tc_1, tc_2, \ldots, tc_n\}$, dont on se servira par la suite. Chaque tc correspond à une correspondance et contient trois attributs (les individus, les assertions, les individus précédents liés) initialement vides.

Instanciation de la classe principale La tâche suivante consiste à instancier la classe principale, pour représenter le document en cours de traitement. Ainsi, un nouvel individu, appelé ci-après l'*instance principale*, instance de la classe principale, est ajouté à O. Dans l'exemple de document de la Figure 3, l'individu $bien1$ est créé avec l'assertion <$bien1, isA, Bien$>[4] ($Bien$ étant la classe principale, représentant un bien immobilier).

3.4.2 Analyse du texte

Cette tâche est détaillée au bas de la Figure 1. Elle analyse les correspondances (via les TC) et vise à ajouter des individus ainsi que des assertions de classe et de propriété en considérant des indicateurs textuels.

Traitement des correspondances de classe Chaque correspondance concernant une classe (sauf la classe principale) est analysée, dans le but de créer un nouvel individu, instance de cette classe. Pour l'exemple en Figure 3, la mise à jour des TC peut être observée (individus et assertions ajoutés) sur les lignes dont le type est « classe » dans le Tableau 1. Ces individus et assertions sont ajoutés à O.
Dans le document, le(s) mot(s) précédent(s) une correspondance sont comparés à une liste de mots-clés exprimant la négation (ex : « pas de »), et la correspondance est ignorée si une trace de négation est trouvée. Par exemple, dans « pas de garage », la correspondance avec « garage » est ignorée, aucune instance de garage n'est créée. Si le mot précédent correspond à un nombre, autant d'individus que le nombre trouvé sont créés. Par exemple, « 2 chambres » engendre la création de deux instances de la classe $Chambre$.

Traitement des correspondances d'individu Ensuite, les TC sont mis à jour pour les correspondances avec des individus de O, en ajoutant les individus concernés. On peut l'observer, pour l'exemple étudié, sur les lignes correspondant au type « individu » dans le Tableau 1.

Traitement des correspondances de propriété Une correspondance avec une propriété doit permettre l'instanciation de cette dernière. Pour ce faire, il faut établir le sujet et l'objet de l'assertion à créer. Ce traitement est décrit dans la suite et est explicité sur des exemples en fin de paragraphe.
Une liste de sujets possibles est considérée. Pour l'instancier, les correspondances candidates sont parcourues, en commençant par celle qui précède la correspondance de propriété en cours de traitement jusqu'au début de la phrase considérée. Dès qu'un tc candidat possède un/des individus qui appartiennent au domaine de la propriété, ces individus constituent la liste des sujets possibles. Pour les objets, tout dépend du type de la propriété à instancier. Dans le cas d'une propriété objet, on considère la correspondance qui suit directement la mention de la propriété. Si une telle correspondance existe, tous les individus de son tc qui sont dans le co-domaine de la propriété sont considérés comme des objets possibles. Dans le cas d'une propriété typée, il existe plusieurs possibilités :
- Si la correspondance vient d'une unité, alors le mot précédent est pris. Dans l'exemple, « chambre de 15 m^2 » a une correspondance entre « m^2 » et *surfaceEnM2* (qui a comme unité m^2). La propriété est instanciée avec la valeur 15.
- Si la correspondance vient d'une expression d'unité, alors la partie qui correspond à la variable est considérée. Par exemple, « Honoraires : 4% » a une correspondance avec la

4. Si un document présente des correspondances avec la classe principale, alors leurs tc sont mis à jour avec l'individu et l'assertion créés. Ce n'est pas le cas dans l'exemple car « bien » ne fait pas partie du texte.

TABLEAU 1 – Les TC de l'exemple après traitements des correspondances de classe, d'individu et de propriété

Correspondance	**Type**	**Individus**	**Assertions**	**Ind. préc.**
Cormelles-le-Royal	individu	Cormelles-le-Royal		
maison	classe	maison1	<maison1, isA, Maison>	
à	propriété objet			
min	propriété typée	distance1	<distance1, isA, Distance><distance1, minVoiture, 15>	
centre-ville	individu	centre-ville		
Caen	individu	Caen		
à	propriété objet			
min à pied	propriété typée	~~distance1~~ distance2	~~<distance1, minPied, 3>~~ <distance2, isA, Distance><distance2, minPied, 3>	
commerces	individu	commerces		
écoles	individu	écoles		
m^2	propriété typée	maison1	<maison1, surfaceEnM2, 110>	
terrain	classe	terrain1	<terrain1, isA, Terrain>	
m^2	propriété typée	terrain1	<terrain1, surface, 400>	
rez-de-chaussée	individu	RDC		
cuisine	classe	cuisine1	<cuisine1, isA, Cuisine>	
équipée	propriété typée	cuisine1	<cuisine1, équipé, true>	
salon	classe	salon1	<salon1, isA, Salon>	
cheminée	classe	cheminée1	<cheminée1, isA, Cheminée>	
exposé	propriété typée	salon1	<salon1, exposition, sud-ouest>	
chambre	classe	chambre1	<chambre1, isA, Bedroom>	
m^2	propriété typée	chambre1	<chambre1, surfaceEnM2, 15>	
Premier étage	individu	premierEtage		
chambres	classe	chambre2 chambre3	<chambre2, isA, Chambre> <chambre3, isA, Chambre>	
sdb	classe	salleDeBain1	<salleDeBain1, isA, SalleDeBain>	
Proche des	propriété objet	bien1	<bien1, estProcheDe, transports_en_commun>	
transports en commun	individu	transports_en_commun		bien1
Terrain	classe	terrain2	<terrain2, isA, Terrain>	
Honoraires : 4%	propriété typée	bien1	<bien1, pourcentageHonoraires, 4>	

propriété *pourcentageHonoraires* (via l'expression d'unité *Honoraires : xxx%*), qui est instanciée avec la valeur 4.
- Si la propriété est booléenne, alors on regarde s'il y a une trace de négation avant. La même liste de mots-clés de négation que dans le traitement des correspondances de classe est utilisée. Par exemple, la propriété booléenne *aménageable* est instanciée avec true pour « le grenier est aménageable » et false pour « le grenier n'est pas aménageable ».
- Si le co-domaine de la propriété est une liste de choix, c'est-à-dire une expression utilisant *owl*:*oneOf*, le(s) mot(s) suivant(s) dans le texte sont comparés aux choix possibles. Par exemple, si la propriété *exposition* a pour valeurs possibles {nord, sud, est, ouest}, alors « exposition sud », conduit à une instanciation avec la valeur « sud ».
- Dans les autres cas, le mot suivant est pris. Par exemple, le texte « construit en 2005 » instancie la propriété *anneeDeConstruction* avec la valeur 2005.

Pour chaque sujet et objet possible, une assertion de la propriété considérée est ajoutée, à condition que celle-ci ne crée pas d'incohérence dans O. Le tc est mis à jour : la ou les nouvelles assertions et leur(s) individu(s) sujet(s) sont ajoutés respectivement aux attributs *assertions* et *individus*. Le tc représentant l'objet est également mis à jour : le sujet de l'assertion est ajouté dans son attribut *individus précédents liés* (attribut exploité dans la suite). Si aucune assertion ne peut être ajoutée, alors le processus est répété avec une nouvelle instance du domaine de la propriété comme sujet. Une fois que toutes les correspondances de propriétés ont été traitées, tous les individus de O qui sont reconnus comme équivalents sont fusionnés, et TC est mis à jour en fonction de cette fusion.

Le Tableau 1 (cf. les lignes dont le type est une propriété) montre le traitement des correspondances de propriété sur l'exemple. Tout d'abord, la mention « à » fait référence à la propriété objet `seSitueA` dont le co-domaine est une commune. Elle est suivie d'une valeur numérique et non d'une correspondance avec une commune. Par conséquent, l'ensemble des objets possibles est vide et la propriété ne peut pas être instanciée. La correspondance sur « min » fait référence à la propriété `distanceMinVoiture` associant une distance à une valeur numérique en minutes. Elle porte sur une unité (« min » est une unité associée à cette propriété), on considère donc le mot précédent comme objet : 15. Il n'y a pas d'instance de la classe `Distance`, l'ensemble des sujets possibles est donc initialement vide, et on considère une nouvelle instance de `Distance` ($distance1$[5]). Pour la correspondance sur « min à pied », le processus est le même, sauf que l'ensemble des sujets

5. Les nouveaux individus sont nommés en fonction de la classe dont ils sont des instances (par exemple, $distance1$ et $distance2$ pour la classe $Distance$). Si le domaine à instancier est une expression de classe, alors les nouveaux individus sont nommés $indiv1$, $indiv2$, etc.

possibles considère $distance1$ puisqu'il s'agit d'un individu résultant d'un tc avant celui que l'on traite, issu de la même phrase, et dans le domaine de la propriété. On tente donc d'ajouter une assertion <$distance1, minAPied, 3$> mais celle-ci est incohérente (car cette distance représente déjà 15 min en voiture). Un nouvel individu $distance2$ est créé pour le sujet. On peut observer dans le tableau toutes les assertions créées. Notons que, comme la correspondance portant sur « Proche des » conduit à l'assertion <$bien1, estProcheDe, transports\_en\_commun$>, l'instance $bien1$ est ajoutée comme individu précédent lié dans le tc de « transports en commun ».

Initialisation des raccrochabilités Dans un contexte descriptif, les verbes ne sont pas très significatifs (cf. Section 2). Il est donc très probable d'avoir manqué, à ce stade, des assertions de propriétés. Le reste de l'algorithme de peuplement vise à ajouter celles-ci. Le défi ici est d'ajouter les assertions manquantes (ce qui augmenterait le rappel) sans en ajouter trop d'erronées (ce qui diminuerait la précision). Dans ce contexte, nous introduisons un ensemble appelé *les raccrochabilités* (Rac). Son initialisation consiste à créer toutes les raccrochabilités $Rac(i)$, pour chaque individu i des TC. Un élément de $Rac(i)$ est un tuple *(propriété, expression de co-domaine)*, tel que i est raccrochable (via la propriété) à un individu appartenant à l'expression de co-domaine. En d'autres termes, pour un individu i, on cherche chaque propriété $prop$ pour laquelle i peut être un sujet. Si i peut être sujet de $prop$, on cherche l'expression de co-domaine à laquelle doit nécessairement appartenir un objet obj d'une éventuelle assertion <i,$prop$,obj>. Pour chaque individu i, l'ensemble des raccrochabilités $Rac(i)$ est automatiquement construit.

Dans l'exemple, prenons l'instance principale $bien1$, qui appartient au domaine de la propriété $contient$: une raccrochabilité est établie. Or, $bien1$ est une instance de la classe $Bien$, sous-classe de l'expression `contient only PartieDeBien`. Ainsi, l'expression de co-domaine associée à $bien1$ et $contient$ est l'intersection du co-domaine de $contient$ (c'est-à-dire `PartieDeBien or PièceDeMaison`) et de la classe $PartieDeBien$ (issue de la définition avec `only`). Le tuple ($contient$, $PartieDeBien$) est donc une raccrochabilité pour $bien1$. Cela signifie que $bien1$ ne pourra être le sujet d'une assertion de $contient$ qu'avec des instances de $PartieDeBien$.

Les raccrochabilités $Rac(i)$ sont exploitées dans les étapes suivantes pour trouver la propriété la plus adaptée pour relier un sujet i avec un objet j. L'ensemble $Rac(i)$ sera parcouru jusqu'à trouver une raccrochabilité telle que j appartienne à l'expression de co-domaine de cette dernière.

$Rac(i)$ est un ensemble trié. Dans le cas où plusieurs propriétés seraient candidates, nous voulons trouver la meilleure, ce qui n'est pas une tâche évidente. Nous avons choisi de donner la priorité à la spécificité. Les premiers éléments sont les raccrochabilités dont le co-domaine est le plus spécifique, puis celles dont le domaine de la propriété est le plus spécifique. Sinon, le tri est arbitraire, en fonction de l'URI de la propriété. Par exemple, si une instance de $Bien$ a pour raccrochabilité l_1 = ($seSitueA$, $Commune$) et l_2 = ($estProcheDe$, $Lieu$) tels que $Commune$ est une sous-classe de $Lieu$, alors ils seront triés dans l'ordre $l_1 < l_2$. Ainsi, si une assertion doit être ajoutée entre $bien1$ et une instance de commune, la propriété choisie sera $seSitueA$ et non $estProcheDe$, puisque le co-domaine de l_1 est plus spécifique que celui de l_2. Dans l'exemple, $Rac(bien1) = \{(seSitueA, Commune), (estProcheDe, Lieu), (seSitueADistance, Distance), (contient, PartieDeBien)\}$.

Traitement des correspondances suivantes Cette étape consiste à vérifier s'il est possible de lier le ou les individus résultant d'une correspondance avec celui ou ceux de la correspondance suivante dans le texte, provenant de la même phrase. Des assertions de propriété sont ajoutées lorsque cela est possible. Les propriétés n-aires sont prises en compte. Une succession de correspondances impliquant respectivement des individus a, b et c peut conduire à des assertions de propriétés du type <*a*,...,*b*> et <*a*,...,*c*>. Ici, le but est de lier a et b, qui sont issus de correspondances consécutives dans le texte, mais aussi a et c, qui ne le sont pas. Pour pouvoir le faire, on exploite les individus précédents liés des TC.

L'idée générale est de regarder si un individu ($sujet$), issu de la correspondance examinée, peut être le sujet d'une assertion ayant pour objet un individu de la correspondance suivante ($objet$), si celle-ci est dans la même phrase. Si la correspondance suivante n'a pas d'individu associé, alors celle d'après est considérée et ainsi de suite. Si $sujet$ et $objet$ ne sont pas déjà reliés entre eux, il peut manquer une assertion. Pour choisir la meilleure propriété possible entre ces deux individus, on considère l'ensemble trié $Rac(sujet)$ et prend la première propriété d'une raccrochabilité telle que $objet$ est dans l'expression de co-domaine de cette dernière et telle qu'elle n'ajoute aucune incohérence à O. L'assertion est ajoutée au tc de $sujet$, et $sujet$ est également ajouté comme un individu précédent lié du tc de l'objet. Si aucune assertion n'est possible entre les individus de

TABLEAU 2 – Les TC de l'exemple étudié après le traitement des correspondances suivantes

Mention	**Individus**	**Assertions**	**Ind. préc.**
Cormelles	Cormelles		
maison	maison1	<maison1, isA, Maison>	
à			
min	distance1	<dist1, isA, Distance> <dist1, minVoiture, 15> <dist1, distDuPI, centre-ville> (1)	
centre-ville	centre-ville	<dist1, distDeLaVillle, Caen> (2)	dist1 (1)
Caen	Caen		dist1 (2)
à			
min à pied	distance2	<dist2, isA, Distance> <dist2, minPied, 3> <dist2, distDuPI, commerces> (3)	
commerces	commerces	<dist2, distDuPI, écoles> (4)	dist2 (3)
écoles	écoles		dist2 (4)
...	...	...	...
salon	salon1	<salon1, isA, Salon> <salon1, aPrElmt, cheminée1> (5)	
cheminée	cheminée1	<cheminée1, isA, Cheminée>	salon1 (5)
exposé	salon1	<salon1, exposition, sud-ouest>	
...	...	...	...

deux correspondances consécutives, alors nous essayons de faire une assertion avec les individus précédents liés comme sujet, afin de faciliter le peuplement des propriétés n-aires. Une fois que tout le texte est examiné, les individus de O reconnus comme équivalents par un moteur d'inférence sont fusionnés, et TC est mis à jour en fonction de cette fusion. Le Tableau 2 montre les TC de l'exemple après cette étape. D'abord, on essaie de lier *Cormelles* à *maison1* (impossible), puis *maison1* à *distance1* (impossible) et ainsi de suite. On peut relier *distance1* à *centre-ville* via la propriété *distanceDuPointDIntérêt* (1). L'assertion est ajoutée dans le tc du sujet (*distance1* portant sur la mention « min »). Le tc suivant (sur la mention « centre-ville ») est mis à jour avec l'individu précédent lié *distance1*. Ensuite, lorsqu'on essaie de lier le tc sur « centre-ville » avec le suivant (sur « Caen »), nous ne pouvons pas lier les individus associés (*centre-ville* et *Caen*). Néanmoins, nous pouvons lier l'individu précédemment lié (*distance1*) avec *Caen*. C'est ce qui est fait dans (2). Et ainsi de suite, on obtient (3), (4) et (5).

Traitement des individus sans prédécesseurs La dernière étape de l'analyse textuelle consiste à traiter les individus sans prédécesseurs. En effet, chaque document décrit une instance de la classe principale et on s'attend à ce que cette instance soit le point de départ des assertions de propriétés. Par conséquent, il semble assez intuitif de penser que chaque individu considéré, sauf l'instance principale, doit être l'objet d'au moins une assertion. L'objectif est donc de trouver des sujets et des propriétés possibles pour les individus (sauf l'instance principale) n'ayant aucun prédécesseur, c'est-à-dire, n'étant pas l'objet d'une assertion de propriété. Pour minimiser le risque de relier des individus qui n'ont rien à voir entre eux, on se concentre uniquement sur les individus résultant de correspondances d'une même phrase. Cela signifie que, pour chaque individu sans prédécesseur ($objet$) d'une phrase, on essaie de lui relier un autre individu de cette phrase ($sujet$), afin d'obtenir l'assertion <$sujet$, $prop$, $objet$>, où la propriété $prop$ choisie est la « meilleure » au sens des raccrochabilités. Enfin, les individus de O qui sont reconnus comme équivalents sont fusionnés, et TC est mis à jour en fonction de cette fusion. La Figure 4 montre les individus (nœuds) et assertions (arêtes) ajoutés avant cette étape pour l'exemple étudié. Les individus sans prédécesseurs (sauf l'instance principale $bien1$) sont grisés. Le Tableau 3 détaille le traitement effectué à cette étape. Chaque ligne correspond à une phrase. Les individus sans prédécesseurs sont en italique. Pour chacun d'eux, on cherche s'il est possible d'ajouter une assertion ayant pour sujet un individu de la même phrase. Par exemple, pour $Cormelles$, on cherche si on peut ajouter des assertions <$maison1$,...,$Cormelles$>, <$dist1$,...,$Cormelles$>, etc. Les seules possibilités pour la première phrase sont données dans la dernière colonne mais elles ne sont pas ajoutées car elles sont incohérentes par rapport à O. En effet, $dist1$ et $dist2$ concernent déjà d'autres lieux et ne peuvent pas concerner $Cormelles$. On répète ce principe pour chaque phrase, et obtient les assertions mentionnées dans le Tableau 3, qui sont ajoutées dans O et TC.

TABLEAU 3 – Traitement des individus sans prédécesseurs

#	Individus	Assertions ajoutées
1	*Cormelles*, *maison1*, *dist1*, centre-ville, Caen, *dist2*, commerces, écoles, *terrain1*	~~<dist1, distDeLaVille, *Cormelles*>~~ ~~<dist2, distDeLaVille, *Cormelles*>~~ non ajoutées car incohérentes
2	*RDC*, *cuisine1*, *salon1*, *chambre1*	<cuisine1, seTrouveEtage, *RDC*> <salon1, seTrouveEtage, *RDC*> <chambre1, seTrouveEtage, *RDC*>
3	1^{er}*Etage*, *chambre2*, *chambre3*, *sdb1*	<chambre2, seTrouveEtage, 1^{er}*Et.*> <chambre3, seTrouveEtage, 1^{er}*Et.*> <sdb1, seTrouveEtage, 1^{er}*Etage*>
4	bien1, transports	
5	*terrain2*	
6	bien1	

3.4.3 Analyse basée sur les connaissances

La dernière tâche de l'algorithme de peuplement est basée sur une analyse des connaissances de O. Elle exploite O, les TC et les raccrochabilités. Idéalement, à partir de l'instance principale, tous les individus du document devraient être atteignables. L'objectif est d'ajouter des assertions de propriété objet manquantes pour obtenir un graphe connexe, dont le point de départ serait l'instance principale. La Figure 5 (haut gauche) montre, entre autres, l'ensemble des individus (nœuds) et des assertions de propriété objet (arêtes) de l'exemple étudié. À partir de l'instance principale $bien1$, seul $transports\_en\_commun$ est atteignable. Les individus et assertions sont mis dans des lots, chaque individu étant dans exactement un lot. Chaque lot est créé à partir d'un individu i, et contient tous les individus qui sont atteignables depuis i (directement ou via une séquence d'assertions). Le premier lot créé est appelé le *lot principal*. Il est composé de l'instance principale et de tous les individus accessibles depuis elle. Ensuite, les individus des assertions restantes sont considérés. On prend celui qui a le moins de prédécesseurs ; l'ordre alphabétique des URI est choisi en cas d'égalité. Son lot est créé, et le processus continue jusqu'à ce que chaque assertion ait été traitée. Enfin, chaque individu restant est respectivement placé dans un nouveau lot. Dans l'exemple étudié, la première division en lots est représentée dans la Figure 5 en haut à gauche (lots encadrés). Le *lot principal* est composé de $bien1$ et $transports$. Ensuite, parmi les assertions restantes, celle

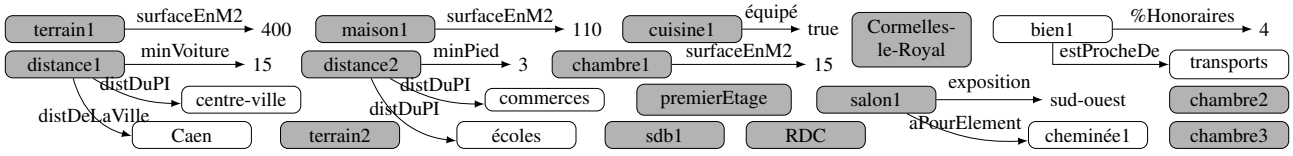


FIGURE 4 – Individus et assertions de propriété de l'exemple étudié, avant le traitement des individus sans prédécesseurs

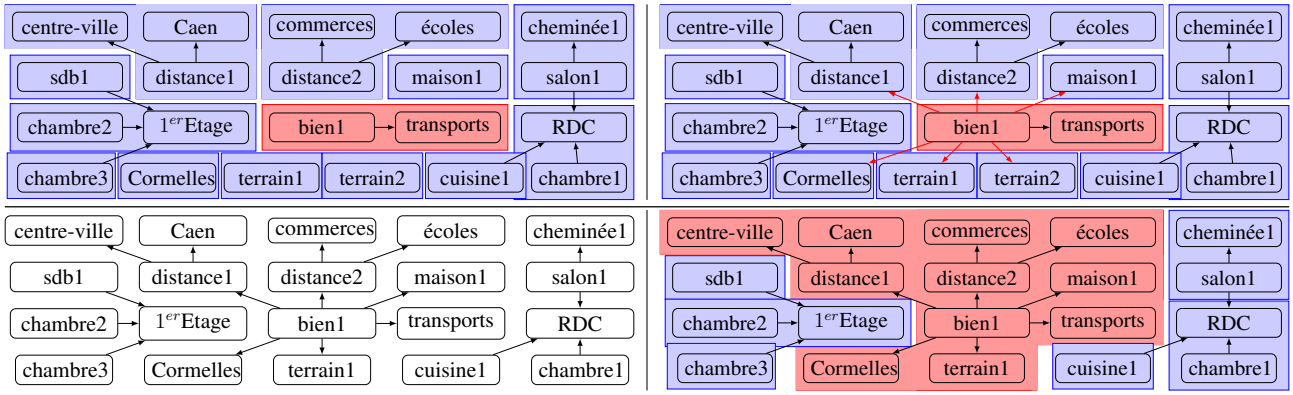


FIGURE 5 – Analyse sur les connaissances : première division en lots (en haut à gauche), premiers ajouts d'assertions (en haut à droite), fusion des individus équivalents (en bas à gauche), seconde division en lots (en bas à droite)

qui a le moins de prédécesseurs (première dans l'ordre alphabétique) est $chambre1$. Le lot suivant est donc constitué de $chambre1$ et RDC. Ainsi de suite, les lots sont formés.

L'objectif est d'accéder à tous les individus à partir de l'instance principale. Ainsi, une fois que tous les lots sont construits, on cherche à lier l'instance principale à un individu de chacun des autres lots. Afin de limiter les erreurs, le lien ne se fait qu'avec un seul individu d'un même lot. Les individus d'un lot sont triés par ordre croissant du nombre de prédécesseurs (puis par ordre alphabétique). Pour chaque lot, on tente de créer une assertion entre l'instance principale (comme sujet) et un individu du lot (comme objet) en respectant le tri, et on s'arrête dès qu'une assertion de propriété peut être établie. Dans l'exemple étudié, $distance2$ est placé en premier dans le lot $\{distance2, commerces, écoles\}$ car il n'a pas de prédécesseurs. Pour traiter ce lot, on regarde d'abord si l'instance principale $bien1$ peut être liée à $distance2$, en utilisant les raccrochabilités. C'est le cas via la propriété $seSitueADistance$. On fait ceci pour chaque lot. Cela permet d'ajouter six assertions, visibles sur la Figure 5 en haut à droite.

Ensuite, les individus équivalents sont fusionnés, et TC est mis à jour en fonction de cette fusion. Dans l'exemple d'ontologie, un bien ne peut contenir qu'un seul terrain. Pour un moteur d'inférence, $terrain1$ et $terrain2$ sont équivalents. Ils sont fusionnés dans $terrain1$ (Figure 5 en bas à gauche).

Le même processus (création de lots, ajouts d'assertions, fusion) est répété, mais en considérant comme sujets tous les individus qui sont à une distance 1 de l'instance principale, c'est-à-dire tous les individus pour lesquels il existe une assertion entre la classe principale et eux. En d'autres termes, la distance d'un individu i peut être définie comme le nombre minimal d'arêtes entre l'instance principale et i. Cette distance est incrémentée progressivement. Nous nous arrêtons soit lorsque le lot principal contient tous les individus, soit lorsque nous avons déjà tenté de relier chaque individu du lot principal. Dans l'exemple, la division en lots est ré-appliquée, conduisant aux lots de la Figure 5 en bas à droite. Le lot principal est plus grand que la première fois, et il n'y a que six autres lots. On cherche maintenant des assertions dont le sujet serait les éléments du lot principal qui sont à une distance de 1 de la classe principale ($Cormelles$, $dist1$, $dist2$, $maison1$, $terrain1$ et $transports$). L'algorithme permet d'obtenir six assertions : <$maison1$, $contient$, $sdb1$/$chambre1$/$chambre2$/$chambre3$/$cuisine1$/$salon1$>. Ainsi, la nouvelle division ne donne plus qu'un seul lot ; autrement dit, tous les éléments sont accessibles à partir de l'instance principale. Le processus est donc arrêté.

L'exemple déroulé est une illustration où KOnPoTe fonctionne bien. Cependant, l'algorithme peut générer des assertions erronées ou oublier des assertions. La section suivante détaille quelques résultats d'évaluation.

4 Expérimentations et évaluation

Cette section présente nos expérimentations sur un corpus d'annonces de vente de maisons. Le protocole expérimental et les résultats obtenus sont discutés. L'approche est implémentée en Java et utilise :

— OWL API [7] pour gérer l'ontologie ;
— Stanford NLP [14] pour découper les textes en phrases ;
— deux lemmatiseurs français [6] : celui d'Ahmet Aker [7] (noté $Aker$) et TreeTagger [8] [17] (noté TT) ;
— le moteur d'inférence Openllet reasoner [9] (Pellet).

4.1 Protocole expérimental

L'approche KOnPoTe a été testée sur un corpus extrait d'un site web [10]. Il contient 78 annonces, en français, annotées comme des ventes d'une maison à Caen. Les informations structurées sur les annonces ont été extraites au format XML, mais nous nous concentrons uniquement sur leur description textuelle. L'ontologie, construite manuellement, décrit le domaine des ventes de maisons, et respecte les contraintes mentionnées dans la Section 3.1. Elle contient initialement quelques individus génériques, tels que *double vitrage*, *transports publics*, etc., ainsi que

6. Des post-traitements ont été implémentés sur les résultats des tokens des lemmatiseurs lors de la considération des mots précédents ou suivants dans le texte. Par exemple, *10 000* devient *10000*, *3 . 4* devient *3.4*, etc.
7. http://staffwww.dcs.shef.ac.uk/people/A.Aker/activityNLPProjects.html
8. https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
9. https://github.com/Galigator/openllet
10. https://www.lecoindelimmo.com/

des entités nommées correspondant à des noms de villes ou de villages. À grande échelle, il faudrait importer une liste importante de villes, mais pour cette expérimentation, nous avons décidé de ne représenter que celles mentionnées dans le corpus. Un Gold Standard (GS) a été construit. Il s'agit de l'ontologie initiale alimentée manuellement avec des assertions représentant les descriptions du corpus. Nous avons appliqué l'approche sur l'ontologie initiale et chaque description du corpus, et nous avons cherché à comparer notre ontologie résultante avec le GS, en calculant la précision (*Pré.*), le rappel (*Rap.*) et la F-mesure (*F-m.*).

$$\textit{Pré.} = \frac{VP}{VP+FP} \quad \textit{Rap.} = \frac{VP}{VP+FN} \quad \textit{F-m.} = \frac{2 \times \textit{Pré.} \times \textit{Rap.}}{\textit{Pré.} + \textit{Rap.}}$$

Afin de calculer ces mesures, nous définissons les assertions de propriété comme des vrais positifs (VP), des faux positifs (FP) ou des faux négatifs (FN). Un VP est une assertion présente à la fois dans le GS et dans l'ontologie obtenue. Un FP est une assertion présente dans l'ontologie obtenue mais absente du GS. Un FN est une assertion présente dans le GS mais absente de l'ontologie obtenue. Pour être le plus juste possible dans nos résultats, nous avons fixé trois règles. Tout d'abord, (1) les assertions de classe ne sont pas prises en compte car elles sont souvent redondantes avec les assertions de propriété. Par exemple, avec l'assertion <$distance1$, $distanceDeLaVille$, $Caen$>, on sait via la définition du domaine de $distanceDeLaVille$ que $distance1$ appartient à la classe $Distance$. De plus, le véritable enjeu de notre problématique se situe au niveau des assertions de propriété, bien plus compliquées à obtenir, les assertions de classe étant le plus souvent issues des correspondances de classe. (2) Les assertions déduites par un moteur d'inférences sont prises en compte, sinon, la comparaison n'aurait pas de sens. Par exemple, si le GS contient l'assertion <a, $prop$, b> et que l'ontologie résultant de notre approche contient <b, $prop^{-1}$, a>, $prop^{-1}$ étant la propriété inverse de $prop$; alors nous devons réaliser que ces deux assertions signifient la même chose. (3) On ignore les propriétés qui nous conduisent à compter plusieurs fois un même élément. Par exemple, s'il existe une propriété et son inverse, un moteur d'inférences traduit une assertion de l'une en une assertion de l'autre. On se retrouve avec deux assertions équivalentes. Dans ce cas, toutes les assertions d'une des deux propriétés sont ignorées. Comme autre exemple, on peut citer le cas où une propriété n'est jamais peuplée par elle-même mais uniquement par le biais de ses sous-propriétés. Dans ce cas, nous ignorons cette propriété, de sorte qu'une assertion (juste ou fausse) ne compte pas deux fois, puisque les assertions inférées sont prises en compte.
Les fichiers expérimentaux sont disponibles [11]. KOnPoTe crée des URIs qui ne sont pas nécessairement les mêmes que ceux du GS. Pour évaluer ces cas-là, des équivalences manuelles sont définies entre les individus du GS et ceux de l'ontologie résultante représentant la même chose. De plus, l'approche peut générer des individus équivalents, sans détecter qu'ils sont équivalents. Dans ce cas, nous supposons qu'un axiome d'équivalence (owl:sameAs) est manquant et le comptons comme une assertion manquante (un FN).

11. Les fichiers expérimentaux (entrées, sorties pour toutes les approches testées, et GS) sont disponibles sur `https://doi.org/10.5281/zenodo.5776752`. Un fichier zip avec un jar exécutable pour KOnPote avec le lemmatiseur *Aker* est disponible sur `https://alec.users.greyc.fr/research/konpote/`.

4.2 Résultats et discussion

Notre problématique étant éloignée des travaux connexes (cf. Section 2), il n'y a pas d'approches concurrentes existantes avec lesquelles se comparer. Nous évaluons KOnPoTe et analysons l'apport de ses modules principaux. La première référence que nous utilisons, nommée *Baseline*, consiste à traiter uniquement les correspondances de classe, d'individu et de propriété (les 3 premières étapes de l'analyse textuelle). Ensuite, dans *Baseline+suivant*, on ajoute le traitement des correspondances suivantes (et l'initialisation des raccrochabilités). Puis, dans *Analyse text.*, on ajoute le traitement des individus sans prédécesseurs, pour enfin ajouter l'analyse basée sur les connaissances (*KOnPoTe*).

TABLEAU 4 – Résultats de KOnPoTe et de trois baselines

Approche	Précision$_{mac}$	Rappel$_{mac}$	F-mesure$_{mac}$
KOnPoTe$_{Aker}$	**0,9516**	**0,8740**	**0,9079**
KOnPoTe$_{TT}$	0,9496	0,8681	0,9039
Analyse text.$_{Aker}$	0,8989	0,4648	0,5994
Analyse text.$_{TT}$	0,8964	0,4579	0,5929
Baseline+suiv$_{Aker}$	0,8911	0,3138	0,4440
Baseline+suiv$_{TT}$	0,8879	0,3081	0,4377
Baseline$_{Aker}$	0,9234	0,1922	0,3099
Baseline$_{TT}$	0,9230	0,1888	0,3054
Approche	**Précision$_{mic}$**	**Rappel$_{mic}$**	**F-mesure$_{mic}$**
KOnPoTe$_{Aker}$	**0,9465**	**0,8606**	**0,9015**
KOnPoTe$_{TT}$	0,9446	0,8545	0,8973
Analyse text.$_{Aker}$	0,8956	0,4726	0,6188
Analyse text.$_{TT}$	0,8937	0,4662	0,6127
Baseline+suiv$_{Aker}$	0,8741	0,3085	0,4561
Baseline+suiv$_{TT}$	0,8732	0,3036	0,4505
Baseline$_{Aker}$	0,9135	0,1926	0,3182
Baseline$_{TT}$	0,9138	0,1892	0,3135

Le Tableau 4 montre les résultats. Les approches sont testées avec les deux lemmatiseurs ($Aker$ et TT). Chaque métrique est calculée de manière macroscopique (mac) et microscopique (mic). Le macro-calcul est la moyenne des métriques pour chaque annonce (chaque annonce a le même poids), tandis que le micro-calcul considère la somme de tous les VP, FP, FN (chaque assertion a le même poids).
Le lemmatiseur *Aker* est plus performant que *TreeTagger*, mais la différence est relativement faible. Les modules ajoutés ont une bonne contribution, puisque chacun permet un gain relativement élevé de F-mesure. Tout d'abord, la baseline, composée des traitements de base, donne un score relativement élevé en précision ($> 0,9$) mais un faible score en rappel ($< 0,2$). Ainsi, la plupart des assertions sont correctes, mais beaucoup d'assertions sont manquantes. L'ajout des modules permet d'ajouter des assertions, qui ont plutôt tendance à être correctes. En effet, au fur et à mesure que les modules sont ajoutés, le rappel augmente sans trop de perte en précision. Plus précisément, le traitement des correspondances suivantes ajoute un peu de

bruit (petite perte de précision) mais augmente le rappel de moitié (de $\sim 0,2$ à $\sim 0,3$). Le traitement des individus sans prédécesseurs augmente également le rappel de moitié (de $\sim 0,3$ à $\sim 0,45$), sans diminuer la précision (en l'augmentant légèrement). Enfin, l'analyse basée sur les connaissances apporte une contribution considérable. La précision et le rappel augmentent, ce qui entraîne une augmentation de moitié de la F-mesure (de $\sim 0,6$ à $\sim 0,9$). Ces trois modules sont donc essentiels : ils insèrent beaucoup d'assertions manquantes sans ajouter trop d'assertions erronées.

5 Conclusion et perspectives

Cet article présente KOnPoTe, une approche générique, entièrement automatique, pour peupler une ontologie de domaine, à partir de descriptions textuelles d'objets de ce domaine. KOnPoTe est une chaîne de traitements, dont les résultats sont prometteurs sur une première expérimentation. Son algorithme est basé uniquement sur le contexte du problème (descriptions textuelles d'objets) et non sur des règles linguistiques propres à un domaine.
Dans un travail futur, nous expérimenterons l'approche sur de nouveaux domaines : d'autres types d'annonces (comme les ventes de bateaux) et des descriptions diverses (hôtels, restaurants, incidents, etc.). Bien sûr, cela est chronophage : constitution du corpus, de l'ontologie, du gold standard, ainsi que des équivalences potentielles entre le gold standard et l'ontologie en sortie de KOnPoTe, et des liens d'équivalence potentiellement manquants dans la sortie. Une autre idée est de tester KOnPoTe sur le même domaine et corpus mais avec des ontologies différentes (même terminologie mais choix de représentation différents). Une analyse approfondie d'une telle expérience pourrait conduire à un ensemble de conseils à suivre pour représenter l'ontologie d'entrée. Enfin, notre objectif final est d'utiliser KOnPoTe comme une première étape pour traiter le problème des annotations erronées mentionné dans la Section 1.

Remerciements

Nous remercions Q. Leroy et J.-P. Kotowicz pour leur participation dans l'élaboration de l'ontologie initiale, ainsi qu'E.-A. Carré et M. Gueret pour la constitution du corpus.

Références

[1] Harith Alani et al. Automatic ontology-based knowledge extraction and tailored biography generation from the web. *IEEE Intell Syst*, pages 14–21, 2003.

[2] Ali Ayadi, Ahmed Samet, François de Bertrand de Beuvron, and Cecilia Zanni-Merk. Ontology population with deep learning-based NLP : a case study on the Biomolecular Network Ontology. *Procedia Computer Science*, 159 :572–581, 2019.

[3] Silvana Castano et al. Multimedia Interpretation for Dynamic Ontology Evolution. *Journal of Logic and Computation*, 19(5) :859–897, 09 2008.

[4] Yohann Chasseray, Anne-Marie Barthe-Delanoë, Stéphane Négny, and Jean-Marc Le Lann. A Generic Metamodel for Data Extraction and Generic Ontology Population. *J. Inf. Sci.*, 48(6) :838—856, dec 2022.

[5] Carla Faria, Ivo Serra, and Rosario Girardi. A domain-independent process for automatic ontology population from text. *Science of Computer Programming*, 95 :26 – 43, 2014.

[6] Houssem Gasmi, Jannik Laval, and Abdelaziz Bouras. Cold-start cybersecurity ontology population using information extraction with LSTM. In *CSET'2019*, pages 1–6, Doha, Qatar, October 2019.

[7] Matthew Horridge and Sean Bechhofer. The OWL API : A Java API for Working with OWL 2 Ontologies. In *OWLED*, page 49–58, Aachen, DEU, 2009.

[8] Ian Horrocks et al. SWRL : A Semantic Web Rule Language Combining OWL and RuleML. Technical report, World Wide Web Consortium, 2004.

[9] Vindula Jayawardana et al. Semi-supervised instance population of an ontology using word vector embedding. In *ICTer*. IEEE, sep 2017.

[10] Natthawut Kertkeidkachorn and Ryutaro Ichise. An Automatic Knowledge Graph Creation Framework from Natural Language Text. *IEICE Transactions on Information and Systems*, E101.D(1) :90–98, 2018.

[11] Andreas Korger and Joachim Baumeister. Rule-based Semantic Relation Extraction in Regulatory Documents. In *LWDA*, volume 2993 of *CEUR Workshop Proceedings*, pages 26–37, September 2021.

[12] Mohamed Lubani, Shahrul Azman Mohd. Noah, and Rohana Mahmud. Ontology population : Approaches and design aspects. *Journal of Information Science*, 45 :502 – 515, 2019.

[13] Jawad Makki, Anne-Marie Alquier, and Violaine Prince. Ontology Population via NLP Techniques in Risk Management. *International Journal of Humanities and Social Sciences*, pages 212–217, 2009.

[14] Christopher D. Manning et al. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL System Demonstrations*, pages 55–60, 2014.

[15] Sergio Oramas, Mohamed Sordo, and Luis Espinosa-Anke. A Rule-Based Approach to Extracting Relations from Music Tidbits. In *Int. Conf. on World Wide Web*, pages 661–666, Florence, Italy, 2015.

[16] José Alejandro Reyes-Ortiz. Criminal Event Ontology Population and Enrichment using Patterns Recognition from Text. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(11), 2019.

[17] Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees, 1994.

[18] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer, 2009.

[19] Fabian Suchanek, Georgiana Ifrim, and Gerhard Weikum. LEILA: Learning to Extract Information by Linguistic Analysis. In *Workshop on Ontology Learning and Population*, pages 18–25, Sydney, Aust., 2006.

Annotation sémantique de documents cliniques psychiatriques français fondée sur une ontologie de domaine

O. Aouina[1,2], J. Hilbey[1,2], J. Charlet[2,3]

[1] Sorbonne Université, Paris, France
[2]Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en e-Santé, LIMICS, Paris, France
[3] Assistance Publique-Hôpitaux de Paris, Paris, France

ons.aouina@etu.sorbonne-universite.fr

Résumé

L'établissement d'une chronologie des profils de patients psychiatriques peut répondre à de nombreuses questions précieuses, telles que la façon dont les événements médicaux importants affectent la progression de la psychose chez les patients. Cependant, la majorité des outils d'extraction d'informations textuelles et d'annotation sémantique, ainsi que les ontologies de domaine, ne sont disponibles qu'en anglais et ne peuvent être facilement étendus à d'autres langues, en raison de différences linguistiques fondamentales. Dans cet article, nous décrivons un système d'annotation sémantique fondé sur une ontologie développée dans le cadre de PsyCARE. Notre système a été comparé avec un autre annotateur sémantique ECMT sur un échantillon de 20 comptes rendus d'hospitalisation français et évalué manuellement par deux annotateurs sur un ensemble de 50 comptes rendus d'hospitalisation, montrant des résultats prometteurs.

Mots-clés

Annotation sémantique, GATE, Ontologie, Psychiatrie, TALN

Abstract

Building a timeline of psychiatric patient profiles can answer many valuable questions, such as how important medical events affect the progression of psychosis in patients. However, the majority of text information extraction and semantic annotation tools, as well as domain ontologies, are only available in English and cannot be easily extended to other languages, due to fundamental linguistic differences. In this paper, we describe a semantic annotation system based on an ontology developed in the framework of PsyCARE. Our system was compared with another semantic annotator called ECMT on a sample of 20 French patient discharge summaries and evaluated by two annotators on a set of 50 summaries, showing promising results.

Keywords

Semantic Annotation, GATE, Ontology, Psychiatry, NLP

1 Introduction

La schizophrénie et la psychose chronique comptent parmi les troubles les plus débilitants chez les adolescents et les jeunes adultes et sont associées à des troubles cognitifs, à une moins bonne réussite professionnelle et à une mauvaise qualité de vie. Des études ont montré que plus la durée de la psychose non traitée est longue, plus les résultats de l'intervention sont mauvais, plus le rétablissement et le fonctionnement général sont mauvais et plus l'affaiblissement social à long terme est important [13].

Ce problème est abordé par le projet RHU PsyCARE [1], qui vise à améliorer l'intervention précoce dans la psychose en fournissant des outils pour faciliter l'accès aux soins et offrir des programmes de traitement personnalisés. Dans ce contexte, l'analyse des Comptes Rendus d'Hospitalisation (CRH) peut nous donner l'opportunité d'étudier de nombreuses questions essentielles, telles que l'impact des événements médicaux importants sur la progression de la psychose chez les patients. Ces résumés fournissent des informations sur l'historique du patient (p. ex. liées au début des symptômes ou au début du traitement ou à des antécédents familiaux).

Cependant, extraire de telles informations pour retracer l'historique de la psychose et développer la chronologie est une affaire complexe qui nécessite des corpus soigneusement annotés. Être en mesure d'extraire automatiquement ces informations peut améliorer les soins médicaux et aiderait également la recherche clinique [1].

Dans ce projet, nous proposons une méthode d'annotation sémantique des CRH fondée sur une ontologie développée dans le cadre de PsyCARE. Notre approche permet non seulement d'extraire les entités médicales d'un texte mais aussi de les transformer en connaissances structurées et formalisées.

La reconnaissance d'entités médicales et l'établissement de liens sont des tâches difficiles dans le traitement du langage naturel en français, en particulier dans notre contexte de textes narratifs non structurés en psychiatrie. Cependant, l'utilisation des ontologies aiderait à concevoir des index de données sémantiques qui exploitent les connaissances mé-

1. https://psy-care.fr/

dicales pour améliorer la recherche et la récupération d'informations. Cette proposition s'inscrit, dans le projet PsyCARE, dans une motivation, large, de construire une chronologie complète de la psychose d'un patient à partir de son dossier médical.

2 Contexte

L'annotation automatique de textes consiste à identifier les passages d'un texte (mots simples ou composés) faisant référence à des concepts identifiés dans des terminologies normalisées (p. ex. les concepts UMLS dans le domaine de la santé, les URI de Dbpedia ou d'autres « Linked Open Data » dans d'autres domaines, etc), puis à identifier les modalités (les concepts qui apparaissent précédés d'une négation ou d'un doute), et enfin, idéalement, les liens entre les concepts.

En raison de l'ambiguïté, de la polysémie du langage naturel, et de la richesse des moyens dont il dispose pour exprimer le sens, cette tâche n'est pas aisée. C'est un champ de recherche actif depuis la fin des années 1970 : le Traitement Automatique du Langage naturel (TALN).

Au cours de la dernière décennie, l'expérience accumulée a conduit à l'émergence de plusieurs plateformes d'analyse de texte solides, offrant la possibilité de procéder avec des performances raisonnables à un premier passage d'annotation automatique. Bien qu'il y ait eu de nombreuses recherches sur l'utilisation de techniques d'annotation sémantique, en particulier l'utilisation d'ontologies, pour améliorer la détection et le diagnostic des troubles mentaux dans les textes en langue anglaise, peu d'études ont exploré l'application de ces techniques dans les textes en langue française [3]. Nous citons notamment :

SIFR [1] est un service web accessible à tous permettant à la fois la reconnaissance et la contextualisation de concepts issus de 30 terminologies et ontologies médicales. Le service d'annotation traite les descriptions textuelles, en les associant à des concepts des ontologies biomédicales adéquats, y compris UMLS. Il crée également des annotations en utilisant les connaissances intégrées dans les ontologies, et contextualise les annotations avant de les renvoyer aux utilisateurs dans plusieurs formats. Son installation en local étant très compliquée, il n'a pas été testé.

ECMT (CHU Rouen, France) [17] est un service web disponible sous la forme d'une API web RESTful, qui permet d'annoter automatiquement un texte à l'aide des concepts des principales terminologies de santé disponibles en Français. Il est inspiré de l'algorithme CISMef pour la recherche d'information avec le moteur de recherche Doc'CISMeF et F-MTI [18] qui est un indexeur automatique multiterminologique. L'ECMT est adapté à la langue française. Il s'appuie sur l'algorithme du sac de mots et sur la reconnaissance de motifs pour analyser des CRH, des rapports de procédures ou des résultats de laboratoire qui contiennent des données symboliques (présence ou absence), des données numériques et des unités de mesure[4]. En revanche, ce dernier ne donne pas la possibilité de choisir l'ontologie à utiliser pour le processus d'annotation. L'annotateur supporte des fonctions d'expansion sémantique. Nous avons testé la version disponible localement avec sept terminologies afin de garantir le respect de la confidentialité des données.

GATE (Univ. Sheffield, UK) [8] est un logiciel *open source* développé à l'origine à l'Université de Sheffield. Il s'agit d'un environnement de TALN qui permet d'effectuer diverses tâches d'analyse de texte, telles que l'extraction d'information, la classification de documents, la recherche d'information et la traduction automatique. Il y a notamment des développements prometteurs dans le domaine de l'annotation sémantique, l'accent étant mis sur l'évolutivité dans le contexte des données ouvertes liées (Linked Open Data) [12]. Il a également été largement appliqué au domaine de la santé[5]. Sa conception permet de créer des flux de traitement en associant des modules *processing units* dans une architecture ouverte. GATE met donc l'accent sur la réutilisabilité et le partage des ressources. Sa licence (AGPL) permet de développer d'autres applications, y compris commerciales (sous réserve d'accord de licence). Chacune de ces plateformes a ses spécificités et ses domaines d'excellence. Parmi elles, GATE est celle qui a été appliquée au domaine de la santé, avec des expériences réussies d'utilisation dans un contexte opérationnel, par exemple au South London and Maudsley University Hospital [6], ou comme spin-off du projet européen Khresmoi/KConnect (outils d'extraction de données sémantiques distribués par la PME bulgare Ontotext). C'est donc cette plateforme qui a été retenue pour notre travail d'annotation.

3 Méthodes

3.1 Jeu de données sur la psychiatrie

Les documents cliniques utilisés dans ce travail sont issus du projet français PsyCARE. Il s'agit d'une compilation d'environ 8000 CRH couvrant une période de dix ans, ce qui représente un volume d'environ 3 500 000 mots. Ces CRH proviennent du Groupe Hospitalier Universitaire Psychiatrie et Neurosciences de Paris. Ils sont semi-standardisés, en format Word et ont été pseudo-anonymisés au préalable, en remplaçant tous les noms, dates, lieux, etc. De plus, le diagnostic est indiqué à la fin de chaque document en utilisant la CIM-10 [2]. Ces documents sont rédigés en français et décrivent l'histoire et le contexte social du patient, les médicaments, les détails de l'admission à l'hôpital et les diagnostics psychiatriques actuels et antérieurs.

3.2 Description de l'ontologie de domaine

L'ontologie utilisée dans notre processus est développée dans le cadre de PsyCARE afin d'intégrer les données et de permettre leur annotation sémantique représentant des domaines tels que les aspects cliniques psychiatriques (signes, symptômes, troubles), les médicaments avec leur code ATC, l'imagerie, la biologie, etc. Comme nous l'avons déjà mentionné, notre objectif final est de modéliser les événements cliniques en psychiatrie. D'où la nécessité d'in-

2. Liste de classification médicale de l'OMS `https://icd.who.int/browse10/2019/en`

clure une représentation temporelle des connaissances médicales [11].
À partir de cette ontologie, nous avons reconstruit un schéma d'annotation, auquel nous avons ajouté une branche décrivant la structure des CRH [3] comme indique la figure 1. Cela nous permet de relier les concepts à leur contexte d'occurrence. Par exemple, les médicaments apparaissant dans la section Historique de la maladie ne partagent pas le même contexte que ceux apparaissant dans la section Traitement de sortie.
Il faut noter que la version utilisée et présentée dans l'article n'est pas finalisée, que ce soit pour la clinique ou pour la description des sections.

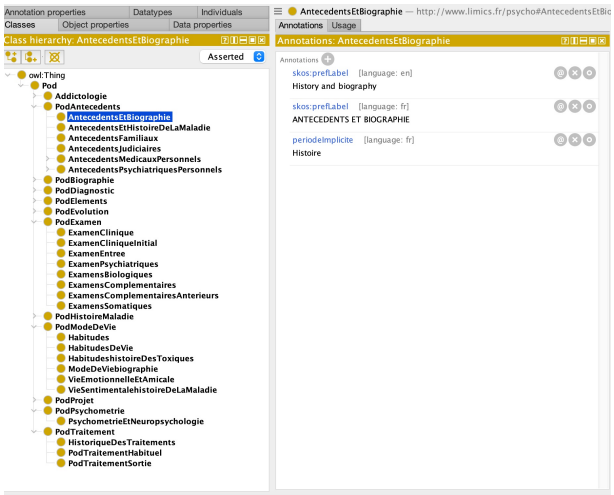


FIGURE 1 – Branche de l'ontologie utilisée pour décrire la structure des CRH des patients.

3.3 Annotation sémantique

GATE fournit différents composants pour l'extraction d'informations sémantiques. Nous avons réutilisé des composants précédemment développés par le projet OnBaSsam [19], qui ont été modifiés et améliorés pour s'adapter aux spécificités des rapports liés à la psychiatrie (p. ex. l'identification des sections et l'extraction des syntagmes nominaux). Notre pipeline est composé de plusieurs ressources de traitement qui s'exécutent séquentiellement sur un document donné, comme le montre la figure 2.

3.3.1 Pré-traitement des comptes rendus médicaux

Tout d'abord, la tokenisation des mots est appliquée sur les CRH, puis le découpage des phrases, suivi du marquage des parties du discours (POS Tagging) et de la lemmatisation, et enfin, l'extraction des syntagmes nominaux ou NP-chunking. Pour ce faire, nous avons utilisé Gate Corpus Pipeline, un composant Gate pour le traitement de corpus, auquel nous avons ajouté les Ressources de Traitement (RT) suivantes : un *tokenizer* français pour la *tokenisation* et le TreeTagger [4] français pour l'annotation des textes avec des informations sur les parties du discours et les lemmes. Ce dernier a été utilisé avec succès pour annoter des textes français. Enfin, pour récupérer les syntagmes nominaux des CRH, OpenNLP [5] est utilisé et adapté à la langue française. Cette étape nous permet de construire une liste de phrases nominales à partir de la sortie du TreeTagger. Dans notre solution, nous supposons que les entités telles que les signes et les symptômes, les maladies, les troubles et les événements cliniques sont des syntagmes nominaux [16]. En ce qui concerne le traitement de l'ontologie, nous suivons les mêmes étapes pour la construction de la liste Gazetteer. Cette liste est constituée des concepts de l'ontologie et de leurs étiquettes (PrefLabels et AltLabels), prétraités selon les étapes présentées dans la figure 2, ainsi que des URI.

3.3.2 Détection des sections

La structure discursive d'un document peut être très utile pour améliorer les outils de recherche d'information. L'identification des sections, par exemple « Histoire de la maladie actuelle » ou « Histoire de la famille », est cruciale dans notre contexte car elle est la clé majeure pour identifier le contexte temporel des passages narratifs. Comme mentionné précédemment, les CRH sont organisés selon une structure taxinomique (Cf. sec. 3.2).
Cette structure n'est pas toujours respectée, certaines sections peuvent être manquantes, fusionnées ou dans un ordre différent, et une même section peut avoir plusieurs titres. Par exemple, pour la section « histoire de la famille », on peut trouver « histoire de la famille », « antécédents familiaux psychiatriques », etc. Par conséquent, nous appliquons les règles JAPE [9] et la correspondance floue des chaînes de caractères sur les termes de la nomenclature des noms de sections précédemment construite pour identifier les limites des sections. Une règle habituelle utilise la distance de Levenshtein qui est une métrique représentant le nombre de changements de caractères entre les mots. Cette règle identifie le début d'une section par l'apparition d'un terme disponible dans la terminologie et la fin de la section avant le début du terme suivant.

3.3.3 Reconnaissance d'entités et établissement des liaisons

Pour cette tâche, nous classons les candidats *NP-chunk* obtenus dans la phase de prétraitement en concepts ontologiques en leur attribuant un URI. Pour ce faire, nous développons un système à base de règles combiné à une classification par chaîne de correspondance floue. Nous définissons le seuil d'acceptation des annotations comme étant proportionnel à la longueur du terme du dictionnaire.
Pour extraire les entités temporelles, nous avons utilisé le plugin TIMEX [6] GATE pour annoter les documents avec des balises TIMEX3 en utilisant la bibliothèque SUTime (Stanford Temporal Tagger). Cela nous a permis d'extraire des dates, des durées, par exemple « depuis environ 1 mois », et des fréquences, par exemple « 2 fois par jour ».

3. Cette branche de l'ontologie est disponible à l'adresse suivante https://github.com/AouinaOns/PartOfDocuments_ontology

4. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger
5. https://github.com/GateNLP/gateplugin-OpenNLP
6. https://github.com/pkourdis/gateplugin-SUTime

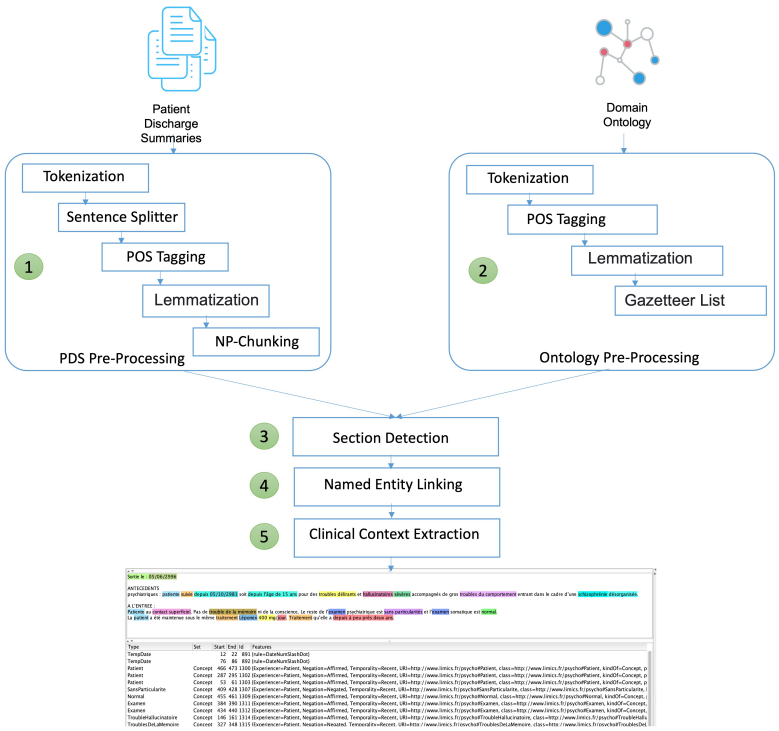


FIGURE 2 – Illustration du processus d'annotation sémantique fondé sur l'ontologie.

3.3.4 Extraction du contexte clinique

En plus d'extraire les entités nommées elles-mêmes, il est nécessaire d'identifier le contexte dans lequel elles apparaissent dans le texte. Les documents sont pré-annotés et mis en correspondance avec les concepts de l'ontologie (Cf. sec. 3.2) (signes, maladies, traitements, symptômes, expressions temporelles, etc.) en utilisant le processus décrit ci-dessus.

Ensuite, l'algorithme français FastContext [14] est appliqué pour identifier le contexte des conditions cliniques annotées dans une phrase. On considère trois contextes : la négation, l'hypothèse et la détermination du sujet, soit le patient, les membres de la famille du patient ou le professionnel de santé.

4 Résultats

4.1 Schéma d'annotation

Nous évaluons d'abord notre approche en analysant manuellement le résultat de la phase d'extraction des entités nommées. Pour ce faire, un schéma d'annotation a été créé pour évaluer les annotations du pipeline. Nous avons utilisé l'outil d'annotation rapide BRAT. Un examen des outils d'annotation Neves et Leser [15] a montré que BRAT était facile à utiliser et pouvait prendre en charge à la fois notre schéma d'annotation et les pré-annotations automatiques. Afin de faciliter l'évaluation manuelle, nos entités extraites ont été regroupées en 10 concepts uniques de premier niveau.

Signe et symptôme. Ce sont des manifestations de troubles mentaux et émotionnels qui peuvent affecter le comportement, la pensée, l'humeur et la perception d'une personne. Ils sont énoncés dans le premier cas par le praticien (signe) et dans le second cas par le patient (symptôme).

Trouble. Cela fait référence à une gamme de conditions de santé mentale qui affecte l'humeur, les émotions, les pensées, le comportement et la perception d'une personne. Les troubles psychiatriques comprennent, par exemple, la dépression, l'anxiété, les troubles bipolaires, les troubles de l'alimentation, les troubles du sommeil et la schizophrénie.

Événement clinique. Les événements cliniques peuvent être des blessures, des accidents, des procédures médicales, des hospitalisations, des interventions chirurgicales, des résultats de tests de laboratoire, des changements dans l'état de santé d'un patient, etc.

Médicament. Ce concept fait référence à une prescription destinée à guérir les maladies ou à soulager les symptômes. On trouve également :

- nom de médicament : Le nom d'un médicament, éventuellement accompagné de sa spécification ;
- dose de médicament : la quantité d'un médicament, y compris le nombre et l'unité.

Situation personnelle. Correspond à tous les aspects de la vie de la personne qui peuvent affecter sa santé et son bien-être. Cela peut inclure des informations sur son environnement familial, social, professionnel, économique, culturel et personnel.

Informations temporelles. Fait référence à des dates, des heures, des durées ou des fréquences dans le texte clinique.

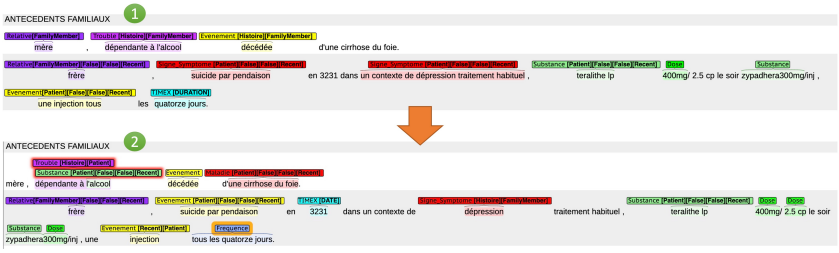


FIGURE 3 – Correction de l'annotation du Pipeline. La partie haute montre la sortie du pipeline et la partie basse, les modifications du correcteur. On constate, par exemple, que l'annotation de « une injection tous les quatorze jours » indique une durée là où il fallait voir une fréquence.

4.2 Évaluation de l'annotation

Nous avons évalué notre approche de deux manières différentes, la première consistant à analyser manuellement la phase de reconnaissance des entités nommées (REN) et d'extraction par rapport à un schéma d'annotation (Cf. sec. 4).

Notre corpus d'évaluation est composé de 50 CRH extraits aléatoirement de l'ensemble de données (4100 phrases, 4213 concepts ontologiques non uniques annotés). Deux personnes ont évalué les annotations et leur contexte, notamment le repérage de la négation, l'hypothétique, la temporalité et la personne impliquée (p. ex. le patient vs un membre de sa famille). La figure 3 montre l'interface de Brat après une correction manuelle des annotations. Nous les avons donc regroupées en 10 concepts uniques de plus haut niveau pour faciliter l'évaluation manuelle. La précision, le rappel et la F-mesure de la phase REN sont présentés dans le tableau 1. La phase NER et l'extraction du contexte des concepts obtiennent une précision globale de 0,9674, un rappel de 0,9780 et une F-mesure de 0,9727.

TABLE 1 – Résultats quantitatifs des évaluations de l'extraction d'entités nommées par les 2 annotateurs avec un accord inter annotateur de 0.88.

		Quantité	Précision	Rappel	F-mes.
Sign Or Symptom		1747	0.9544	0.9503	0.9524
Disease		150	0.9826	0.7635	0.8593
Trouble		459	0.9894	0.9493	0.9690
Clinical Event		1459	0.9744	0.8140	0.8870
Personal Situation		188	0.9895	0.8468	0.9126
Drug	Drug Name	1034	0.8200	0.9805	0.8822
	Drug Dose	650	0.9848	0.9610	0.9727
Temporal Inf.	Date	840	0.9942	0.9709	0.9824
	Duration	529	0.9574	0.9777	0.9674
	Frequency	212	0.9459	0.8373	0.8883

Le deuxième mode d'évaluation consiste à comparer nos performances avec celles d'un autre outil d'annotation sémantique [2], l'ECMT, sur un échantillon de 20 CRH.

Nous avons observé que notre pipeline a extrait le plus grand nombre de concepts : 2384 alors que 1838 ont été extraits avec l'ECMT. Notre pipeline est plus efficace dans l'identification des signes, symptômes et troubles psychiatriques, tels que les délires et le tempérament hyperthymique, ainsi que les informations temporelles, qui sont des informations courantes dans les CRH. Inversement, l'ECMT détecte mieux les maladies et les événements non psychiatriques, tels que la sinusite maxillaire récurrente et l'appendicectomie.

5 Discussion

L'évaluation initiale des annotations sémantiques montre que nous pouvons identifier correctement les concepts de l'ontologie en utilisant GATE et des algorithmes modifiés pour tenir compte des spécificités des textes en français. Ces résultats encourageants peuvent être expliqués par deux facteurs. Premièrement, la richesse du vocabulaire de l'ontologie, notamment pour la composante signes et symptômes psychiatriques. Deuxièmement, la prise en compte de la structuration des CRH, qui pourrait être une faiblesse pour d'autres types de documents psychiatriques. De plus, cette méthode permet de déterminer les limites correctes des entités nommées détectées.

Cependant, il convient de noter que les expressions nominales obtenues automatiquement (*NP-Chunk*) ne sont pas toujours parfaites et peuvent entraîner des erreurs. Bien que nous ayons consacré des efforts considérables à la correction des annotations du corpus, la taille de l'ensemble des données annotées reste modeste par rapport à d'autres travaux.

Une autre limite réside dans le fait que le pipeline présenté dans cet article repose principalement sur des techniques traditionnelles de traitement du langage. Pour l'améliorer, des règles doivent être ajoutées manuellement, ce qui peut être chronophage et ne permet pas toujours de capturer toutes les nuances des CRH. En réponse aux limites de notre approche, une combinaison d'approches fondées sur les règles et l'apprentissage automatique peut être utilisée pour construire des systèmes d'annotation sémantique plus efficaces [10].

6 Conclusion

L'objectif de ce travail est de reconstruire les données structurées des patients à partir des CRH afin de compléter les données des patients dans le projet PsyCARE. Dans cet article, nous avons présenté une première étape qui consiste à annoter sémantiquement les CRH de psychiatrie français. A cette fin, à partir d'un ensemble non structuré, nous avons

pu réaliser une annotation sémantique en utilisant des plugins GATE et des algorithmes que nous avons modifiés pour les adapter à la structure des CRH.
Les prochaines étapes de notre travail consistent à intégrer dans l'ontologie les événements cliniques et les maladies non-psychiatriques les plus fréquemment présents dans les CRH. Ensuite, il sera nécessaire d'identifier les relations entre les concepts. Par exemple, extraire les relations temporelles entre les événements cliniques et psychiatriques et les entités temporelles. Cela permettra de retracer de manière exhaustive la chronologie des événements dans les rapports cliniques, offrant ainsi une représentation plus complète du parcours thérapeutique des patients. Compte tenu de l'état de l'art, cela sera fait avec des algorithmes d'apprentissage automatique [7, 20].

Remerciements

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du Programme d'Investissements d'Avenir portant la référence PsyCARE ANR-18-RHUS- 0014.

Références

[1] A. Tchechmedjiev, A. Abdaoui, V. Emonet, S. Zevio, C. Jonquet. SIFR annotator : ontology-based semantic annotation of French biomedical text and clinical notes. *BMC Bioinformatics.* 2018.

[2] A. Redjdal, J. Bouaud J, J. Gligorov, B. Séroussi. Comparison of MetaMap, cTAKES, SIFR, and ECMT to Annotate Breast Cancer Patient Summaries. Stud Health Technol Inform. 2022 Jun 6.

[3] A. Le Glaz, Y. Haralambous Y, DH. Kim-Dufor, P.Lenca et al. Machine Learning and Natural Language Processing in Mental Health : Systematic Review. J Med Internet Res. 2021 May 4.

[4] C. Cabot, Recherche d'information clinomique dans le Dossier Patient Informatisé : modélisation, implantation et évaluation. Thèse de doctorat, 2017.

[5] G. Gorrell, J. Petrak, and K. Bontcheva. Using Twitter Conventions to Improve #LOD-Based Named Entity Disambiguation. The Semantic Web. *Springer International Publishing*, pages 171–186, Cham, 2015.

[6] G. Perera, M. Broadbent, F. Callard, et al. Cohort profile of the South London and Maudsley NHS Foundation Trust Biomedical Research Centre (SLaM BRC) Case Register : current status and recent enhancement of an Electronic Mental Health Record-derived data resource, *BMJ Open* 2016.

[7] G. Alfattni, N. Peek, and G. Nenadic. Extraction of temporal relations from clinical free text : A systematic review of current approaches. *Journal of biomedical informatics*, 2020.

[8] H. Cunningham, D. Maynard, K. Bontcheva et al. Text processing with GATE (Version 6). Sheffield : GATE, 2011.

[9] H. Cunningham, D. Maynard, and V. Tablan. JAPE : a java annotation patterns engine (second edition). department of computer science, university of sheffield, 2000.

[10] J. Jouffroy, SF. Feldman, I. Lerner,B. Rance, A Burgun, A. Neuraz. Hybrid Deep Learning for Medication-Related Information Extraction From Clinical Texts in French : MedExt Algorithm Development Study. JMIR Med Inform. 2021 Mar 16.

[11] J. Hilbey, X. Aimé, and J. Charlet. Temporal Medical Knowledge Representation Using Ontologies. Studies in Health Technology and Informatics 294 (May 25, 2022) : 337–41.

[12] L. Derczynski, D. Maynard, G. Rizzo et al. Analysis of named entity recognition and linking for tweets, Information Processing & Management, Volume 51, Issue 2, 2015.

[13] L. Souaiby, R. Gaillard R, MO.Krebs . Durée de psychose non traitée : état des lieux et analyse critique [Duration of untreated psychosis : A state-of-the-art review and critical analysis].2016 Aug.

[14] M. Mirzapour, A. Abdaoui, A. Tchechmedjiev et al. French FastContext : A publicly accessible system for detecting negation, temporality and experiencer in French clinical notes, *Journal of Biomedical Informatics*, Volume 117, 2021.

[15] M. Neves, U. Leser, A survey on annotation tools for the biomedical literature, Briefings in Bioinformatics, Volume 15, Issue 2, March 2014.

[16] S. Zhang, N. Elhadad, Unsupervised Biomedical Named Entity Recognition : Experiments with Clinical and Biological Texts. Journal of biomedical informatics. 2013.

[17] S. Pereira, A. Névéol, G. Kerdelhué, E. Serrot, M. Joubert, S. Darmoni. Using multi-terminology indexing for the assignment of MeSH descriptors to health resources in a French online catalogue. AMIA symp. 2008 :586-590.

[18] S. Darmoni, JP. Leroy, M. Douyère et al. Doc'CISMeF : un outil de recherche Internet orienté vers l'enseignement et la formation à distance en médecine.

[19] S. Cardoso, P. Meneton, X. Aimé, V. Meininger, D. Grabli, G. Guezennec, J. Charlet, Use of a modular ontology and a semantic annotation tool to describe the care pathway of patients with amyotrophic lateral sclerosis in a coordination network. *PLoS One*. 2021 Jan.

[20] Y. Luo, Ö. Uzuner, P. Szolovits . Bridging semantics and syntax with graph algorithms-state-of-the-art of extracting biomedical relations. Brief Bioinform. 2017.

Session Posters et Démonstrations

CubicWeb as a Service : un service pour la publication sur le Web de données liées

Fabien Amarger[1], Nicolas Chauvat[1], Élodie Thieblin[1]
[1] Logilab, 104 avenue Blanqui, Paris

prénom.nom@logilab.fr

Résumé

La publication de données RDF, suivant les 5 étoiles du Linked Open Data, devient de plus en plus courante. Ces données sont généralement publiées sous forme de fichiers dump sur des portails, ou rendues interrogeables en SPARQL.
Il est moins commun d'accéder aux données via une interface web et grâce au mécanisme de négociation de contenu du protocole HTTP, probablement car cela demande des compétences techniques supplémentaires pour déployer et maintenir ces formes d'accès.
Dans cet article, nous décrivons nos travaux sur CubicWeb et OWL2YAMS qui, ensemble, permettent de générer, à partir d'une ontologie OWL et de données RDF, une application Web offrant la négociation de contenu, une interface de consultation et une interface de gestion des données.
Nous y ajoutons l'automatisation du déploiement, pour qu'il devienne possible, via un formulaire Web, de partir de données de qualité en RDF et d'obtenir un en quelques clics une application Web complète pour gérer, publier et visualiser ces données.
CubicWeb-as-a-Service a pour but de rendre toujours plus aisée la publication de données sur le Web en respectant les standards du W3C.

Mots-clés

Web de données liées, publication de données, RDF, CubicWeb

1 Introduction

De plus en plus de structures adoptent les principes des données ouvertes et liées (5 étoiles du Linked Open Data) en traduisant leurs données en RDF. Toutefois, la publication de ces données de bonne qualité se fait par un dump RDF (un fichier contenant la totalité des triplets RDF) ou par téléversement des triplets RDF dans un entrepôt RDF interrogeable en SPARQL. La négociation de contenu (API native du Web) est plus coûteuse à mettre en place, car elle requiert de configurer le serveur sur lequel seront servies les données.
Une fois les données en RDF publiées, elles sont généralement plus difficile à modifier directement en RDF que dans l'outil au sein duquel elles ont été produites et qui offre une interface spécifiquement adaptée.
CubicWeb et OWL2YAMS présentés dans des articles précédents[1, 2] permettent de prendre des données en RDF et de créer une application Web de gestion et publication des données, avec négociation de contenu selon le protocole HTTP.
La question de l'hébergement d'une l'application CubicWeb pose un problème pour les personnes et les institutions avec peu de compétences techniques.
CubicWeb-as-a-Service (CWaaS) est un outil en ligne qui automatise la création et l'hébergement d'une application CubicWeb à partir d'une ontologie OWL et des données RDF qui l'instancient.
Cet outil a pour objectif de rendre plus accessible la publication et la gestion de données sur le Web en respectant les standards du W3C.

2 CubicWeb et OWL2YAMS

CubicWeb[1] est un logiciel libre écrit en Python, conçu pour faciliter le développement et le déploiement d'applications qui reprennent les concepts essentiels du Web Sémantique. Avec CubicWeb, il est aisé de gérer et de rendre accessibles des données qui suivent un modèle préalablement défini.
CubicWeb utilise le formalisme YAMS (Yet Another Magic Schema [1]) pour représenter le modèle de données de façon explicite et stocker les données suivant ce modèle dans une base de données relationnelle, en profitant d'une gestion fine des permissions.
CubicWeb permet de configurer l'export RDF des données stockées et de les rendre accessibles via la négociation de contenu du protocole HTTP.
Des interfaces Web de gestion et de consultation des données génériques sont disponibles dans CubicWeb.
CubicWeb est utilisé dans des applications de grande envergure comme Data.BnF [2] ou FranceArchives [3].
OWL2YAMS[2] permet de créer une application CubicWeb à partir d'une ontologie OWL et de la peupler avec des données RDF décrites par ladite ontologie.
À eux deux, ces outils facilitent la création d'une application Web de publication, consultation et gestion des don-

1. https://forge.extranet.logilab.fr/open-source/yams
2. data.bnf.fr/
3. https://francearchives.gouv.fr/

nées à partir de données en RDF, mais ils n'adressent pas la question de l'hébergement.

3 Déploiement automatique

Nous avons appelé notre approche **CubicWeb-as-a-Service** (CWaaS) car elle offre la possibilité de déployer une instance de CubicWeb automatiquement. L'idée principale est de proposer une mini-application Web, dans laquelle téléverser une ontologie OWL et des données en RDF respectant cette ontologie, pour qu'une application CubicWeb soit créée et déployée automatiquement. Ce service est donc à la portée de toutes et tous et permettrait de faciliter l'appropriation des technologies du Web Sémantique en les rendant extrêmement simple d'utilisation.

Sur la figure 1 est présentée l'interface d'accueil de CWaaS, sur laquelle ne figurent que trois champs dans le but de la rendre la plus simple possible.

Le premier champ est pour le nom de l'application (ce nom sera aussi utilisé pour générer l'URL à partir de laquelle l'application sera disponible).

Le deuxième champ permet de téléverser l'ontologie que nous souhaitons utiliser pour la création de l'application CubicWeb (dans cet exemple, nous utilisons l'ontologie SKOS exprimée dans la syntaxe N3).

Le troisième champ permet de téléverser un fichier RDF contenant les données qui seront automatiquement chargées dans cette application une fois qu'elle aura été déployée (dans cet exemple, un thésaurus exprimé en SKOS avec une syntaxe XML).

Une fois que le bouton "Create instance !" est actionné, l'ontologie est interprétée pour générer un cube CubicWeb en utilisant OWL2YAMS comme expliqué dans [2].

Le composant applicatif (cube) produit par OWL2YAMS est ensuite envoyé dans un nouvel entrepôt de code automatiquement créé dans notre forge GitLab [4]. Nous pourrons ensuite pousser plus loin le développement de ce cube pour ajouter des fonctionnalités ou modifier les permissions par exemple (c.f. [1] pour des détails) et en employant pour cela les outils et processus habituels de développement logiciel.

Nous utilisons le système de déploiement continu intégré à GitLab pour construire l'image Docker [5] contenant l'application créée à partir de notre cube, puis pour déployer automatiquement cette image Docker sur un *cluster* Kubernetes [6].

Pour suivre l'évolution de la création de l'image et du déploiement, les états d'avancement des jobs de déploiement continu sont affichés sur l'interface de CWaaS, comme nous pouvons le voir sur la figure 2.

Sur cette même figure, nous pouvons observer plusieurs liens. Tout d'abord le lien vers le dépôt de code de l'application générée. Ensuite, un lien pour vérifier l'état d'avancement des tâches du déploiement continu directement sur l'entrepôt de code. Et enfin le lien vers l'instance de l'application qui sera valide lorsqu'elle sera déployée.

Lorsque l'étape de déploiement est terminée, nous pouvons alors accéder à l'application elle-même, comme nous pouvons le voir sur la figure 3.

Nous observons sur cette figure que l'application déployée a bien intégré les classes de l'ontologie SKOS en chargeant les données (il y a ici 506 instances de `skos:Concept` dans l'application).

L'application est à ce stade disponible, avec les données chargées. Il est possible d'utiliser l'interface d'administration pour modifier les données ou la configuration du serveur. Si une modification est apportée au code dans le dépôt de code, le processus d'intégration continue sera relancé et mettra à jour automatiquement l'application déployée. Il devient donc assez simple, même pour des développeurs ou développeuses, de faire évoluer cette application pour ajouter de nouvelles fonctionnalités, notamment en ajoutant des interfaces personnalisées pour afficher les données de la manière qui convient le mieux.

Dans le cadre du projet CubicWeb, nous sommes en train de travailler sur une nouvelle interface d'administration générique qui s'appuie sur la bibliothèque React-Admin [7], illustrée par la figure 4.

4 Conclusion et perspectives

Nous présentons ici notre approche *CubicWeb-as-a-Service* permettant de créer une instance de CubicWeb et de la déployer très facilement en utilisant une application Web monopage sur laquelle figure un formulaire composé de trois champs. Toute la pile technique utilisée est disponible sous licence libre aux adresses suivantes :

- CubicWeb
  `https://forge.extranet.logilab.fr/cubicweb/cubicweb`
- OWL2YAMS
  `https://forge.extranet.logilab.fr/cubicweb/owl2yams`
- CubicWeb-as-a-Service
  `https://forge.extranet.logilab.fr/cubicweb/cubicweb-as-a-service`

Les autres outils employés sont Docker, Kubernetes, GitLab et GitLab-runner qui sont eux aussi sous licence libre.

Plusieurs évolutions de CWaaS sont prévues pour améliorer son utilisation.

Tout d'abord, nous souhaitons rendre plus robuste OWL2YAMS. Effectivement, comme détaillé dans [2], il y a un écart d'expressivité entre la définition du modèle de données dans CubicWeb (YAMS) et OWL-lite. Nous souhaitons améliorer YAMS et OWL2YAMS pour atteindre un niveau d'expressivité similaire à OWL-lite.

L'interrogation des données d'une instance de CubicWeb se fait avec un langage d'interrogation nommé RQL[1]. Bien que ce langage soit assez proche du SPARQL, il existe quelques différences d'expressivité. Nous avons commencé à étudier la possibilité d'interroger une instance CubicWeb en SPARQL, notamment avec l'outil OnTop[3]. Nous avons commencé à développer un cube qui permet

4. `https://docs.gitlab.com/`
5. `https://www.docker.com/`
6. `https://kubernetes.io/fr/`
7. `https://marmelab.com/react-admin/`

FIGURE 1 – Interface d'accueil de CWaaS

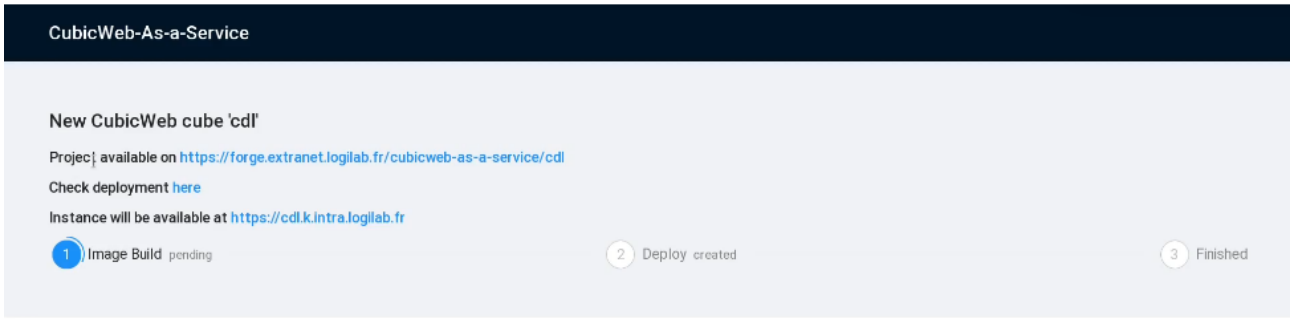


FIGURE 2 – Suivi de l'état d'avancement du déploiement de l'application

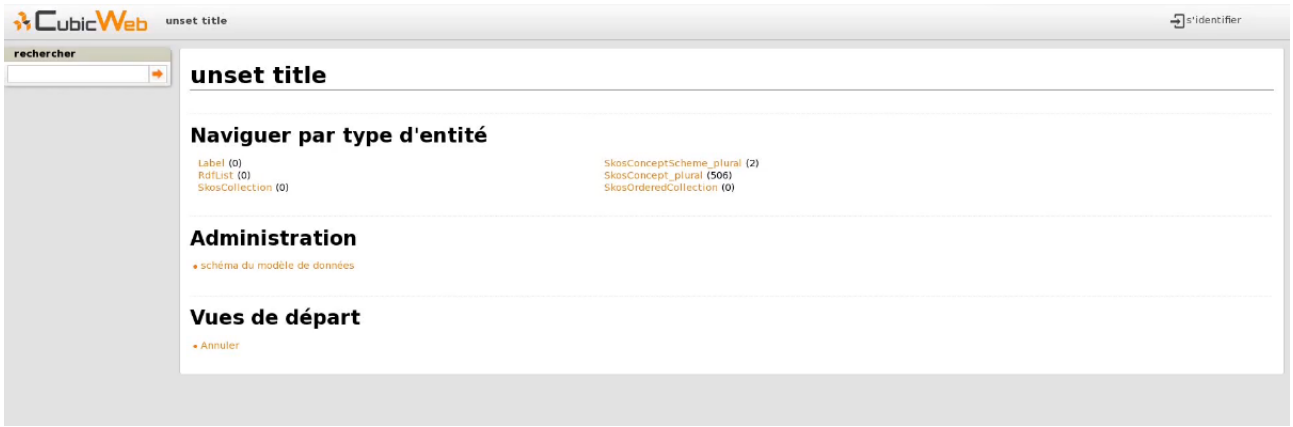


FIGURE 3 – Interface par défaut de CubicWeb

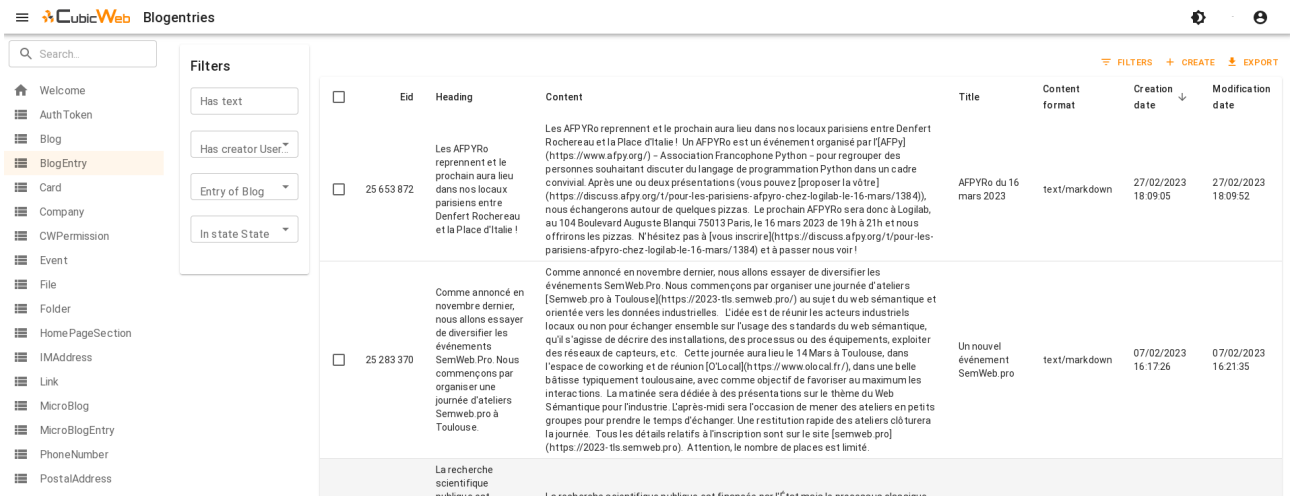


FIGURE 4 – La nouvelle interface d'administration React-Admin

de créer automatiquement la configuration OnTop à partir d'un schéma YAMS (`https://forge.extranet.logilab.fr/cubicweb/cubes/ontop`). Il nous faudrait continuer ces expérimentations pour valider l'hypothèse qu'il est possible d'interroger une instance de CubicWeb en SPARQL.
Enfin, nous avons l'intuition que notre approche concernant CubicWeb, qui permet notamment de faire une séparation claire entre le serveur fournissant les données et les interfaces permettant de les afficher, est très proche de ce qui est proposé dans SOLID [8]. Nous souhaitons approfondir ce lien pour déterminer comment une instance CubicWeb peut interagir avec un pod SOLID, voir comment une instance de CubicWeb pourrait devenir un pod SOLID.

Références

[1] Fabien Amarger, Simon Chabot, Nicolas Chauvat, and Elodie Thiéblin. Cubicweb : vers un outil pour des applications clé en main dans le web sémantique. In *31es Journées francophones d'Ingénierie des Connaissances*, 2020.

[2] Fabien Amarger, Nicolas Chauvat, and Elodie Thiéblin. Owl2yams : créer une application cubicweb à partir d'une ontologie owl. In *Journées francophones d'Ingénierie des Connaissances 2022*, 2022.

[3] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. Ontop : Answering sparql queries over relational databases. *Semantic Web*, 8(3) :471–487, 2017.

8. `https://solidproject.org/`

Classification incrémentale d'objets dans un graphe de connaissances à partir d'informations issues de capteurs

Vincent Beugnet[1,2], Nathalie Pernelle[1], Manel Zarrouk [1], Cyrille Enderli[2], Ludovic Grivault[2]

[1] LIPN, CNRS (UMR 7030), Université Sorbonne Paris Nord, Villetaneuse, France
[2] Thales DMS France SAS, 2 avenue Gay Lussac, Elancourt Cedex, 78851 France

20 mai 2023

Résumé

Représenter et classifier des données incomplètes et évolutives dans un graphe de connaissances est d'importance dans de nombreuses applications. Dans certains contextes, il est possible de compléter les descriptions des individus en faisant appel à des ressources externes comme des capteurs. C'est ainsi le cas d'un théâtre d'opération où un capteur détecte un objet, et où d'autres capteurs peuvent être mobilisés pour compléter ces premières informations. Dans ce papier, nous proposons une méthode de classification incrémentale qui exploite une ontologie de domaine et un score évaluant le pouvoir discriminant d'une propriété afin de choisir un capteur qu'il faudra mobiliser.

Mots-clés

Classification, Ontologie, Graphes de connaissance, Capteurs, Données évolutives, Sélection de propriété

1 Introduction

De nombreux raisonneurs permettent de classifier des individus décrits en RDF automatiquement en se basant sur les descriptions des classes qui ont été formalisées dans une ontologie de domaine. Si l'information décrivant un individu est souvent incomplète et ne permet pas toutjours de le classifier précisément, il existe certains contextes dans lesquels des informations complémentaires peuvent être acquises pour compléter une description. C'est le cas par exemple d'un théâtre d'opérations militaires où de nouveaux objets peuvent être détectés grâce à un ou plusieurs matériels ou capteurs (e.g. radar mobile, capteur optoélectronique...). Dans un tel cadre, il est bien sûr important de classifier précisément l'objet et d'autres capteurs peuvent être mobilisés pour compléter une description initiale afin de classifier l'objet aussi précisément que possible. Cependant, la difficulté provient du fait que tous les capteurs présents sur le théâtre d'opération ne peuvent pas être mobilisés pour enrichir la description d'un objet donné. Leur utilisabilité est évolutive en fonction des différents évènements qui se déroulent dans le même temps.

Dans ce travail, nous utilisons une ontologie de domaine qui réutilise en partie des ontologies de haut niveau existantes permettant de décrire des capteurs et des observations [1] afin de décrire des classes et des propriétés nécessaires pour représenter des données évolutives sur les objets présents sur un théâtre d'opération. Nous proposons un processus de classification incrémental qui s'appuie sur le caractère discriminant d'une propriété numérique ou symbolique afin de minimiser le nombre de capteurs à mobiliser pour classifier précisément un objet. Le fait de se baser sur une ontologie et sur un raisonneur permettra de fournir des éléments d'explication à un opérateur afin qu'il comprenne comment une classe a été sélectionnée avant une possible prise de décision.

2 Etat de l'art

Ontologies définies pour représenter les capteurs et les observations :
De nombreuses ontologies ont été développées pour modéliser des capteurs et les informations qui en sont issues, que ce soit dans le domaine de l'observation environnementale [2] ou géologique [3], de la robotique [4] ou de l'internet des objets [5]. Pour le secteur militaire, l'ontologie Device du projet Marine Metadata Interoperability (MMI) [6] vise à faciliter la recherche d'informations sur les capteurs d'observation océanographique et leur utilisation. Créée plus récemment, l'ontologie créée dans le cadre du Missions & Means Framework (MMF) [7] fait le lien entre plateformes, capteurs et missions pour permettre de choisir une plateforme adaptée à chaque mission.

Depuis 2017, l'ontologie SSN (Semantic Sensor Network) composée du cœur SOSA (Sensor, Observation, Sample and Actuator) [1] fait partie des recommandations du World Wide Web Consortium (W3C) et des implémentations standards de l'Open Geospatial Consortium (OGC). Elle permet de modéliser des capteurs, leurs observations, leurs plateformes et leurs propriétés. Cette ontologie a montré ses usages possibles dans de nombreux domaines et a défini des alignements avec d'autres ontologies telles qu'Observations and Measurements (O&M) [8] de l'OGC et l'ontologie fondationnelle Dolce-Ultralite (DUL) [9].

Classification incrémentale et sélection de caractéristiques : De nombreux travaux en apprentissage se sont intéressés à la classification incrémentale supervisée [10] qui fait habituellement référence aux algorithmes permettant d'entraîner un modèle de manière incrémentale : quand

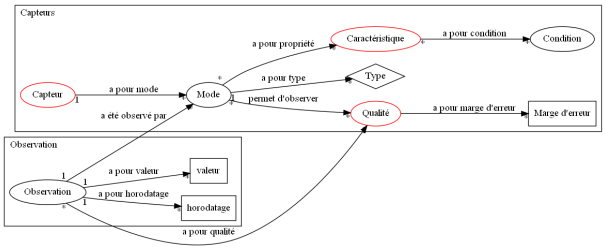


FIGURE 1 – Description des capteurs (extrait)

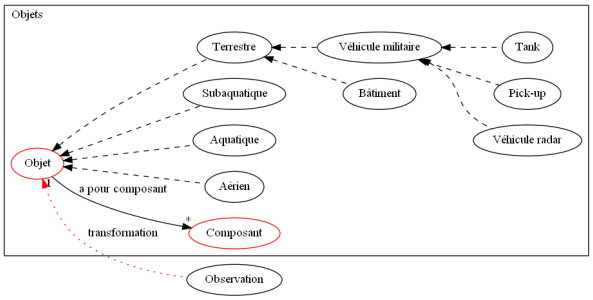


FIGURE 2 – Classes d'objet du domaine (extrait)

de nouvelles données labellisées sont présentées à l'algorithme, le système peut adapter le classifieur sans repartir de zéro. Contrairement à ces approches, nous nous intéressons au cas où les définitions associées aux classes sont statiques tandis que la description d'un objet peut évoluer.
D'autres travaux s'intéressent à la sélection de caractéristiques, qui consiste à déterminer le sous-ensemble de variables les plus pertinentes lors du développement d'un modèle prédictif afin de réduire le temps de calcul du modèle ou d'améliorer ses performances [11]. Dans notre approche il s'agit de sélectionner la meilleure propriété à compléter afin de classifier précisément un objet donné et la propriété la plus discriminante peut varier selon les informations déjà disponibles sur cet objet. Des approches telles que [12], utilisent des axiomes déclarés dans l'ontologie pour collecter dans des resources externes des propriétés discriminantes, telles que des propriétés inverses fonctionnelles, pour inférer des liens d'identité entre les individus qui peuplent une ontologie, mais il ne s'agit pas d'un problème de classification.

A notre connaissance, il n'existe pas de travaux qui se soient intéressés à l'utilisation de SOSA associée à une ontologie de domaine pour définir des stratégies de classification incrémentale permettant de minimiser le nombre de capteurs à utiliser, et de minimiser leur temps d'utilisation.

3 Une Ontologie basée sur SOSA

Dans cette section, nous présentons un extrait de l'ontologie développée au sein de Thales DMS qui est utilisée pour représenter les objets présents sur un théâtre d'opérations ainsi que leurs propriétés. Nous présentons tout d'abord les classes les plus importantes de la partie haute de l'ontologie puis quelques exemples de classes et de propriétés qui serviront d'illustration pour les exemples utilisés lors de la classification.
Un *capteur* est décrit par des modes d'utilisation (classe $Mode$), chacun permettant d'obtenir des valeurs pour un ensemble de propriétés donné (classe $qualite$) avec une certaine marge d'erreur. Une *observation* permet de lier une qualité à une valeur observée. Les capteurs sont également décrits par un ensemble de $caracteristique$s qui précisent par exemple leur portée, ou leur temps d'acquisition.
L'extrait d'ontologie présenté en Figure 2 décrit un sous-ensemble de classes du domaine (e.g. $vehicule\ terrestre$). Les propriétés associées à ces classes vont être utilisées pour classifier les objets observés. Les exemples utilisés pour illustrer notre approche se basent sur les trois classes fictives suivantes :

- La classe $FordRanger$ est une sous-classe de $Pickup$ dont les instances ont une vitesse maximum de 120 km/h, une largeur de 190 cm, une longueur de 510 cm, les armes embarquées possibles sont AE1, AE3.
- La classe $TankIT - ELX$ est une sous-classe de $Tank$ dont les instances ont une vitesse maximum de vitesse maximum est de 100km/h, une largeur de 360 cm, une longueur de 750 cm, et qui peut disposer de deux sortes d'armes embarquées AE1 et AE2.‘
- La classe $VRadarER18$ est une sous-classe de $VehiculeRadar$ dont les instances ont une vitesse maximum de 60km/h, une largeur de 250 cm, une longueur de 960 cm, et n'embarquent pas d'armes.

Un objet est également décrit par un ensemble de propriétés obtenues par des capteurs. Ces informations sont issues d'un processus d'aggrégation sur les données observées (i.e. valeur de qualité et marge d'erreur du capteur). On représente par exemple ainsi la vitesse maximum observée (e.g. le minimum à zero n'est pas informatif), ou encore des intervalles de valeurs pour des propriétés comme la largeur $largeurObsMin$ et la longueur $largeurObsMax$. Ces données seront mises à jour au fur et à mesure du temps (voir sous-section 4.1).
Comme l'objectif de cette ontologie est de classifier de nouveaux objets dont la description observée est évolutive, des contraintes sont ajoutées qui relient les valeurs observées aux valeurs définies pour une classe. Ainsi, si la propriété $Largeur$ est définie pour une classe, cette valeur devra être comprise entre les valeurs de largeur observées minimum $LargeurObsMin$ et maximum $LargeurObsMax$ pour qu'un objet appartienne à cette classe. De même, les armes embarquées observées $typeArmeEmbarquesObs$ devront appartenir à l'ensemble possible d'armes embarquées définies pour la classe.

4 Approche de classification

L'objectif de l'approche développée est de découvrir la classe la plus précise à laquelle appartient un objet compte tenu des informations décrites dans les observations qui

ont été associées à cet objet. Un sous-ensemble des classes d'intérêt est défini par l'objectif de la mission. Quand un capteur va détecter un nouvel objet sur un théâtre d'opérations, une nouvelle instance de la classe $Objet$ est créée qui est décrite par les informations que ce capteur a pu obtenir. Comme tous les capteurs ne sont pas utilisables à un moment donné pour obtenir toutes les informations manquantes concernant un objet, notre approche de classification est incrémentale : à chaque étape il s'agira d'obtenir les classes possibles et de déterminer les propriétés manquantes qui permettraient de réduire au maximum le nombre de classes possibles. Ces étapes seront répétées autant de fois que nécessaire pour classifier un objet ou le rejeter s'il ne fait pas partie des classes définies dans l'objectif de la mission.

4.1 Première étape de classification

La première tentative de classification débute par la création d'un nouvel objet auquel on associe un ensemble d'observations faites sur l'objet lorqu'il est detecté pour la première fois. Sa description est obtenue en utilisant les valeurs obervées et en prenant en compte les marges d'erreur définies $MargeP$ pour la propriété et le capteur. Pour une propriété p ayant une valeur numérique, l'objet sera décrit par un p_{ObsMax} seul, ou un p_{ObsMax} et un p_{ObsMin}. Pour une propriété p ayant une valeur symbolique, l'objet sera décrit par l'ensemble des valeurs différentes obtenues grâce aux observations capteurs.
Exemple (étape 1) : On dispose de 2 observations sur un objet $o1$:
$\{VitesseObs(o1, 50km/h), DistanceObs(o1, 232km)\}$.
Le capteur $ca1$ utilisé est défini par les marges d'erreurs suivantes :
$\{MargeVitesse(ca1, 5km/h), MargeDistance(ca1, 2km)\}$
Un nouvel objet $o1$ est alors créé qui est associé à la description suivante :
$\{vitesseObsMax(o1, 45), distanceObsMin(o1, 230km), distanceMaxObs(o1, 234km)\}$

La valeur de $vitesseObsMax$ est obtenue en prenant le minimum de l'intervalle obtenu pour la vitesse la plus élevée observée en tenant compte de la marge d'erreur, ceci afin de ne rejeter aucune classe à cause cette erreur.

La description obtenue est utilisée pour classifier l'objet dans l'ontologie et donc obtenir via un raisonneur l'ensemble des classes candidates $C(o1)$ les plus spécifiques.

Exemple (étape 2) : Pour l'étape 1, l'ensemble de classes obtenues $C(o1)$ est :
$\{FordRanger, VRadarER18, TankIT - ELX\}$
On considère que la classification est incomplète tant que cet ensemble C ne se réduit pas à l'ensemble vide ou à une seule classe feuille de l'ontologie. On ne conserve une classe possible pour un objet que si elle est égale ou plus générale qu'une classe définie par l'objectif de mission.

4.2 Choix de la propriété la plus discriminante

Si la classification est incomplète, on cherche à la compléter en un minimum d'étapes. Pour cela, nous choisissons d'utiliser la propriété la plus discriminante possible pour les classes auxquelles l'objet peut appartenir. On considère l'ensemble des propriétés non renseignées ou évolutives pour $o1$ qui appartiennent à la description d'au moins une des classes de $C(o1)$. Quand une propriété p ne fait pas partie des propriétés décrivant une classe, nous considérons que cette classe dispose d'une valeur par défaut pour cette propriété (i.e. 0 pour les propriétés numériques et $\emptyset$ pour les valeurs symboliques).
Nous considérons un score sd_s pour évaluer le caractère discriminant d'une propriété symbolique p pour un objet o. Ce score se base sur la mesure de similarité Jaccard définie pour comparer les ensembles de valeurs de p notés $val(p)$.
$sd_s(p, o) = 1 - \frac{\sum_{(ci,cj)\in C(o)\times C(o)} jaccard(val(p_{ci}), val(p_{cj}))}{|(ci,cj)|}$

Le score sd_n évalue le caractère discriminant d'une propriété numérique p pour un objet o en se basant sur l'écart type σ des différentes valeurs de p après normalisation min-max dans les classes $C(o)$.

$sd_n(p, o) = \sigma(\frac{val(p)-min(val(p))}{max(val(p))-min(val(p)}$)

Exemple (étape 3) : Dans l'étape 2, les attributs possibles non renseignés ou évolutifs pour $o1$ sont : $\{typeArmesEmbarqueesObs, vitesseObsMax, largeurObsMax, longueurObsMax\}$.
Compte tenu de la description des 3 classes, on obtient :
$sd_s(typeArmesEmbarquees) = 0,890$
$sd_n(vitesseObsMax) = 0.509$
$sd_n(largeurObsMax) = 0.507$
$sd_n(longueurObsMax) = 0.500$

On établit ainsi un ordre au sein des attributs, en les classant du plus discriminant au moins discriminant.

4.3 Choix du capteur

Une fois la liste ordonnée des propriétés discriminantes établie, on recherche tous les capteurs capables de fournir une valeur pour les n premiers attributs, (a) en fonction des caractéristiques de l'objet (e.g. certains capteurs ne peuvent être utilisés si l'objet est à trop grande distance), et (b) en respectant certaines contraintes telles que la discrétion. Ces filtrages successifs permettent d'aboutir à une liste ordonnée de capteurs capables d'obtenir à l'instant t une information pour une propriété, dans laquelle les capteurs permettant d'obtenir les propriétés les plus discriminantes sont en tête de liste.
Exemple (étape 4) : Dans l'étape 3, l'attribut le plus discriminant est $typeArmesEmbarquees$ pour l'ensemble des classes possibles. On détermine qu'il est possible d'utiliser les capteurs $ca1$, $ca2$ ou $ca3$ pour donner une valeur à cette propriété. Seulement, on sait que pour utiliser $ca1$, la cible doit être fixe, ce qui n'est pas le cas d'$o1$. On retire donc

$ca1$ des capteurs possibles. De plus, il y a une contrainte de distance à respecter pour utiliser $ca3$ que l'objet ne respecte pas, on va donc choisir d'utiliser $ca2$ pour obtenir l'information manquante. Si aucun des trois capteurs n'avait été utilisable, le même filtrage serait appliqué au deuxième attribut le plus discriminant.

4.4 Etape de classification $n+1$ et Conditions d'arrêt

Après un appel au capteur sélectionné, on obtient une information supplémentaire sur l'objet qui devrait réduire le nombre de classes candidates et donc nous aider à le classifier. On reprend la première étape de classification avec les informations dont nous disposions, auxquelles on ajoute la nouvelle information observée obtenue après agrégation.

Exemple (étape 5) : A partir des informations obtenues par le capteur $ca2$ à l'étape 4, on peut enrichir les connaissances sur l'objet en observant l'absence d'arme embarquée ; l'objet $o1$ sera décrit par :
$\{vitesseMaxObs(o1, 50km.h^{-1}),$
$distanceMaxObs(o1, 234km),$
$distanceMinObs(o1, 230km),$
$typeArmesEmbarquees(o1, \emptyset)\}$
$o1$ sera alors classé comme instance de $VRadarER18$ que l'on ne peut plus spécialiser.

Il existe trois conditions d'arrêt pour la classification :
-Si les informations dont nous disposons ne correspondent à aucun objet connu (ie. l'étape de classification ne renvoie aucune classe), cela signifie que l'objet observé n'est pas représenté dans la base de connaissances,
-Si les informations dont nous disposons correspondent à au moins une classe de l'ontologie, mais celles-ci ne correspondent à aucune classe définie dans l'objectif de mission, ou que l'objet est hors de portée des capteurs, il est inutile de continuer l'identification,
-Si l'objet observé a été parfaitement identifié (i.e. une seule classe possible, feuille de l'ontologie),
Dans tous les cas, les informations, i.e. règles et valeurs seront sauvegardées pour une vérification par un opérateur, mais également dans l'objectif éventuel d'ajouter une classe manquante à la base de connaissance.

5 Conclusion

Ce papier présente une première approche de classification incrémentale d'objets dont la description est évolutive. L'approche est basée sur une ontologie de domaine dont les classes de domaine ont été enrichies par des propriétés et des contraintes liant les valeurs observées aux valeurs définies pour une classe. Le choix du capteur est basé sur un score évaluant le caractère discriminant d'une propriété qui permet de déterminer les prochains capteurs à utiliser. Dans des travaux futurs, nous montrerons sur un ensemble de données synthétiques et réelles que cette stratégie permet de minimiser le nombre de capteurs à mobiliser et de comparer l'utilisation des scores sd_s et sd_n proposés ici à d'autres stratégies de sélection des meilleures propriétés.

Références

[1] A. Haller, K. Janowicza, S. J. D. Cox, M. Lefrançois, K. Taylor, D. L. Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, "The modular ssn ontology : A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation," *Semantic Web, vol. 10, no. 1, pp. 9-32, 2019*, 2018.

[2] A. J. G. Gray, J. Sadler, O. Kit, K. Kyzirakos, M. Karpathiotakis, J.-P. Calbimonte, K. Page, R. Garcia-Castro, A. Frazer, I. Galpin, A. A. A. Fernandes, N. W. Paton, O. Corcho, M. Koubarakis, D. D. Roure, K. Martinez, and A. Gomez-Perez, "A semantic sensor web for environmental decision support applications," *Sensors*, 2011.

[3] S. Chien, D. Tran, J. Doubleday, A. Davies, S. Kedar, F. Webb, G. Rabideau, D. Mandl, S. Frye, W. Song, B. Shirazi, and R. Lahusen, "A multi-agent space, in-situ volcano sensorweb," 2010.

[4] E. Prestes, S. R. Fiorini, and J. Carbonera, "Core ontology for robotics and automation," 2014.

[5] G. M. Honti and J. Abonyi, "A review of semantic sensor technologies in internet of things architectures," 2019.

[6] C. Rueda, N. Galbraith, R. A. Morris, L. E. Bermudez, R. A. Arko, and J. Graybeal, "The mmi device ontology : Enabling sensor integration," 2010.

[7] "The missions & means framework ontology : Matching military assets to mission objectives," 2016.

[8] S. J. D. Cox, "Ontology for observations and sampling features, with alignments to existing models," 2015.

[9] S. Borgo, R. Ferrario, A. Gangemi, N. Guarino, C. Masolo, D. Porello, E. M. Sanfilippo, and L. Vieu, "Dolce : A descriptive ontology for linguistic and cognitive engineering," 2006.

[10] C. Salperwyck and V. Lemaire, "Classification incrémentale supervisée : un panel introductif," in *Apprentissage Artificiel et Fouille de Données, AAFD 2010, Université Paris 13, Institut Galilée, series = RNTI, volume = A-5, pages = 121–148, publisher = Hermann-Éditions, year = 2010*.

[11] J. M. L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016.

[12] M. Al-Bakri, M. Atencia, S. Lalande, and M. Rousset, "Inferring same-as facts from linked data : An iterative import-by-query approach," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, USA*, pp. 9–15, AAAI Press, 2015.

Tutoriel sur DLinker : Un outil rapide de découverte d'entités similaires entre deux graphes de connaissances

Bill. Gates Happi Happi[1,4], Géraud Fokou Pelap[2], Danai Symeonidou[3], Pierre Larmande [1,4]

[1] LIRMM, Department of Computer Science, University of Montpellier, Montpellier, 34095, France
[2] URIFIA, Department of Computer Science, University of Dschang, Dschang, 00237, Cameroon
[3] Mistea, INRAE, Montpellier, 34060, Montpellier
[4] Diade, IRD, Montpellier, 34394, France

20 mars 2023

Résumé

Dans ce tutoriel, nous allons vous présenter notre logiciel de recherche d'entités similaires dans les graphes de connaissances, qui a été conçu pour aider les utilisateurs à identifier des informations partageant des données communes ou équivalentes. Notre logiciel offre une flexibilité d'utilisation dans le processus d'appariement, avec une gestion des paramètres simplifié pour la rigueur dans le calcul de l'appariement. Au cours de la démonstration, nous allons présenter en 5 étapes, comment notre logiciel peut aider à réaliser la tâche d'appariement d'entités. Notre logiciel est facile à installer et à utiliser, avec des fonctionnalités puissantes pour optimiser la recherche d'appariement de graphes de connaissances. Nous sommes convaincus que notre logiciel peut aider à gagner en efficacité et en productivité dans la gestion des processus d'appariement de toutes tailles et complexités.

Mots-clés

Graphe de connaissances, alignement d'entités, liage de données.

Abstract

In this tutorial, we will introduce you to our software for entity linking in knowledge graphs, which was designed to help users to identified shared information or equivalent data accross datasets. Our software offers flexibility of use in the matching process, with a simplified parameter management to control the matching. During the demo, we will show in five steps how our software can help to perform the task of entity matching. Our software is easy to install and use, with powerful features to optimize knowledge graph matching. We are confident that our software will help the users to gain efficiency and productivity in managing the matching processes of all sizes and complexities.

Keywords

Knowledge graph, entity matching, data linking.

1 Plan du tutoriel

1.1 Description du tutoriel

Détection rapide et précise d'entités similaires entre deux graphes de connaissances complémentaires. Considérons deux entreprises A et B ayant des bases de données de trajectoires (chemins) empruntées par leurs clients. Sur la base d'un nouveau contrat de prestation de services qui ferait baisser simultanément leurs coûts de transports, ils décident d'identifier leurs trajectoires communes séparées, régulièrement utilisées dans l'optique d'avoir un seul service de transport. Notre tutoriel considère que les deux bases de données ont été converties en graphe de donnée au format RDF et aura pour objectif de démontrer la capacité de l'outil DLinker à retrouver rapidement des chemins similaires entre ces deux graphes de données complémentaires.

1.2 Points du tutoriel

Le tutoriel suppose que les utilisateurs ont déjà l'outil python3.8, pip et git installés dans leurs ordinateurs.

1.2.1 Installation des dépendances de l'outil

Dans la section readme du répertoire github du projet sera détaillé la procédure d'installation des dépendances. En résumé, elle consiste à installer les bibliothèques de fonctions permettant de faciliter la navigation dans les graphes de données au format RDF.

1.2.2 Clonage du projet depuis github

Dans ce point, nous exécuterons la commande ci-dessous en une seule ligne :

```
1 git clone
2 https://github.com/BillGates98/DLinker
```

1.2.3 Téléchargement des jeux de données

le lien de téléchargement est disponible vers ce lien :
`https://users.ics.forth.gr/~jsaveta/.index.php?dir=OAEI_HOBBIT_LinkDiscovery/Spatial/Spaten_LinesTOLines/CONTAINS/`

1.2.4 Copie des données vers le répertoire dédié

Une fois rendus à l'adresse précédente, nous verrons 3 liens. Le premier lien nommé **GoldStandards** [1] mène vers le fichier de vérités pour évaluer la précision des résultats obtenus par l'outil. Le deuxième lien nommé **SourceDatasets** [2] contient la première source de données à lier. Le troisième lien nommé **TargetDatasets** [3] contient la seconde source de données à lier à la précédente.

1.2.5 Exécution de l'outil

Elle consistera à lancer deux commandes depuis un script shell (disponible sur le répertoire github [4] de l'outil ainsi que la sortie attendue), qui réaliseront respectivement les tâches suivantes :

1. Alignera les prédicats similaires entre les deux graphes de données ;
2. Alignera les instances ou entités à partir de la sortie précédente.

1.2.6 Sortie attentue

L'outil fournira un graphe, en d'autres terme un ensemble de triplets ayant pour prédicat **owl :sameAs**. Les sujets seront des entités du fichier source et les objets seront les entités du fichier cible.

1.3 Caractérisation du public

Le tutoriel est destiné à toute personne ayant des bases élémentaires en informatique et capable d'installer des outils de programmation informatique. Ces personnes devraient savoir exécuter un script python avec des paramètres.

1.4 Raisons ou Intérêts du tutoriel

Toute personne qui souhaiterait retrouver des entités similaires entre deux graphes de connaissances partageant des entités équivalentes. Nous rajoutons aussi que toute base de données converties au format RDF pourraient très bien subir ce processus. Les participants à PFIA peuvent :

1. Apprendre à utiliser cet outil pour formaliser leurs connaissances et les rendre plus précises et plus faciles à partager avec d'autres chercheurs ;
2. Améliorer la récupération de l'information à partir de bases de données et de sources d'information en ligne. Ils peuvent apprendre à utiliser des ontologies pour organiser les informations de manière cohérente et à développer des systèmes de recherche plus précis et plus rapides ;
3. Rendre leurs données plus accessibles à un public plus large ;
4. En utilisant des ontologies et des langages de représentation des connaissances, ils peuvent faciliter la compréhension des concepts et des résultats de leurs recherches par des personnes n'appartenant pas à leur domaine ;
5. Améliorer leur compréhension du monde.

Orateur

Nom : HAPPI HAPPI Bill Gates
Affiliation : laboratoire IRD unité Diade équipe Ceres
Numéro de téléphone : +33784898009
Adresse électronique : bill.happi@ird.fr
Expérience en Web Sémantique : 3 ans
Exemple de travail disponible : « https ://ceur-ws.org/Vol-3324/oaei22_paper6.pdf ».

Remerciements

Les auteurs remercient le plateau bio-informatique i-trop de l'IRD UMR DIADE pour l'accès à son infrastructure. Les auteurs remercient également l'ANR et le projet DIG-AI pour son soutien financier.

1. https ://users.ics.forth.gr/~jsaveta/.index.php ?dir=OAEI_HOBBIT_LinkDiscovery/Spatial/Spaten_LinesTOLines/CONTAINS/GoldStandards
2. https ://users.ics.forth.gr/~jsaveta/.index.php ?dir=OAEI_HOBBIT_LinkDiscovery/Spatial/Spaten_LinesTOLines/CONTAINS/SourceDatasets
3. https ://users.ics.forth.gr/~jsaveta/.index.php ?dir=OAEI_HOBBIT_LinkDiscovery/Spatial/Spaten_LinesTOLines/CONTAINS/TargetDatasets
4. https ://github.com/BillGates98/DLinker

Gestion de connaissances en maintenance aéronautique à l'aide d'une ontologie

Ba-Huy Tran, Thi-Bich-Ngoc Hoang, Marzieh Mozafari
Capgemini Engineering

prenom.nom@capgemini.com

Résumé

Au cours des vingt dernières années, le rôle de la maintenance dans les entreprises est devenu de plus en plus important tant sur le plan technologique qu'économique. Pourtant, les besoins des acteurs de la maintenance et des utilisateurs évoluent dans le temps et ne peuvent être satisfaits par les services actuellement fournis par les systèmes informatiques parce que ces prestations s'appuient sur des connaissances initialement formalisées mais qui ne sont ni homogènes ni systématiquement mises à jour. Cet article présente notre effort réalisé dans le cadre d'un projet R&I qui vise à gérer les connaissances en maintenance aéronautique et renforcer l'exploitation de ces connaissances à l'aide d'un modèle ontologique.

Mots-clés

Ontologie, maintenance aéronautique, procédure de maintenance.

Abstract

Over the past twenty years, the role of maintenance in companies has become increasingly important both technologically and economically. However, the needs of maintenance actors and users change over time and cannot be satisfied by the services currently provided by systems because these services are based on knowledge that was initially formalized but is not homogeneous nor systematically updated. This article presents our effort carried out within an R&I project which aims to manage knowledge on aircraft maintenance and to reinforce the exploitation of this knowledge using an ontological model.

Keywords

Ontology, aircraft maintenance, maintenance procedure.

1 Introduction

La maintenance industrielle est une fonction métier stratégique. Elle peut être définie comme l'ensemble des actions de dépannage et de réparation, de réglage, de révision, de contrôle et de vérification des équipements matériels voire immatériels. Au cours des vingt dernières années, le rôle de la maintenance dans les entreprises est devenu de plus en plus important tant sur le plan technologique qu'économique. Qu'il s'agisse des dépenses de maintenance industrielle ou du personnel dédié, le secteur de la maintenance affiche une augmentation significative sur tous les points. En France, les dépenses annuelles de maintenance avoisinent les 18 milliards d'euros et nécessitent 70 000 emplois [1].

D'ailleurs, les besoins des acteurs de la maintenance et des utilisateurs évoluent dans le temps et ne peuvent être satisfaits par les services actuellement fournis par les systèmes informatiques. En effet, ces prestations s'appuient sur des connaissances initialement formalisées mais qui ne sont pas systématiquement mises à jour. Ainsi les prestations proposées ne peuvent pas tenir compte de l'aspect dynamique des connaissances qui évoluent au fil du temps. Il est indispensable que les plateformes de maintenance doivent renforcer l'exploitation des connaissances de maintenance en développant la standardisation des informations et des connaissances en termes de compréhension, d'interprétation et de partage, améliorant ainsi l'interopérabilité sémantique.

Au sein de Capgemini Engineering, notre projet R&I IREPAIR a été lancé pour répondre aux besoins de maintenance industrielle, plus particulièrement la maintenance aéronautique. En effet, il existe un fort besoin sur l'analyse et la gestion des jeux de données sur la maintenance, notamment les manuels de maintenance et les demandes de maintenance ou réparation. Ils proviennent de différentes sources, élaborées par différentes équipes. L'ontologie est considérée comme un bon moyen permettant l'interopérabilité de ces sources hétérogènes. Dans cet article, nous décrivons notre travail en cours sur le développement d'une ontologie aidant à représenter la connaissance à gérer ainsi que les pistes d'exploitation pour la base de connaissances construite.

2 Développement de l'ontologie

Nous avons choisi une approche incrémentale basée sur la « technologie sémantique » qui vise à hybrider connaissances formelles et linguistiques dans les technologies sémantiques [10]. Notre approche est illustrée par la Figure 1.

Initialisation de l'ontologie. Tout d'abord, l'ontologie est initialisée avec les concepts clés représentant la connaissance du domaine et des procédures de maintenance comme décrites dans les sections suivantes. Cette étape s'appuie principalement sur les outils disponibles pour extraire et

1. https ://metclasse.fr/les-chiffres-cles-maintenance/

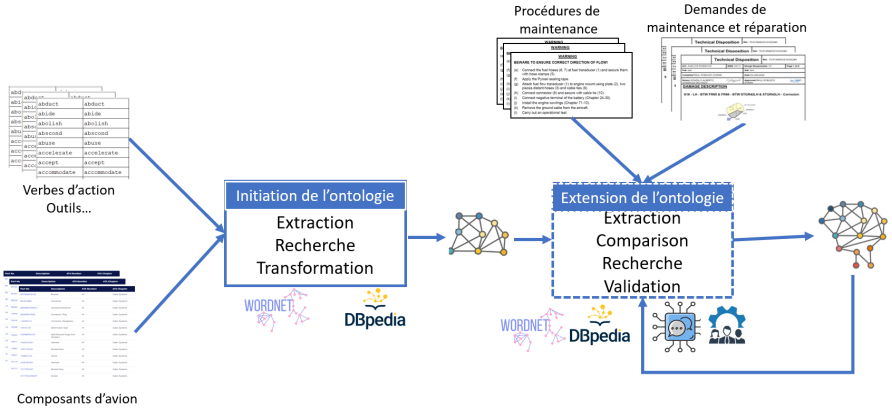


FIGURE 1 – Approche incrémentale proposée pour construire l'ontologie

transformer les données structurées en concepts ontologiques. Cette méthode permet d'avoir une ontologie de base au plus vite à partir des données structurées.
L'ontologie initialisée est aussi servie comme une référence pour la connaissance linguistique, comme décrite dans [10]. Ainsi, nous pourrons améliorer le modèle linguistique entraîné qui est utilisé dans l'étape suivante.

Extension de l'ontologie. Il est tout à fait évident que l'ontologie initialisée ne couvre pas tous les concepts nécessaires, surtout les composants d'avion. Afin de pallier ce problème, nous nous appuyons sur la solution d'apprentissage ontologique [9] qui est supervisée par des experts. Le TALN est appliqué pour extraire et annoter [2], de manière automatique, des informations pertinentes issues des textes non structurés. Elles seront ensuite nettoyées et comparées avec les concepts disponibles dans l'ontologie. Enfin, les concepts nouveaux sont importés dans l'ontologie après avoir été validés par les experts.

3 Représentation des procédures de maintenances

Nous examinons tout d'abord la capacité d'analyser et représenter des procédures de maintenance aéronautique qui sont généralement décrites dans des documents appelés *Aircraft Maintenance Manual* (ou AMM, un exemple du document se trouve par ici [3]). Afin d'aboutir à un modèle ontologique adapté, nous considérons qu'une procédure de maintenance est une procédure particulière pour faire quelque chose impliquant une ou plusieurs étapes ou opérations. En modélisation des processus métier, le modèle *Business Process Model and Notation* (BPMN) est une norme largement utilisée. Les spécifications ont donné lieu à la construction d'ontologies, telles que l'ontologie BPMN [1], ou BBO [6]. En modélisation des procédures industrielles, [2] a introduit une ontologie pour la maintenance industrielle, et [3] a introduit une ontologie pour le processus de fabrication. Par ailleurs, il existe plusieurs travaux inspirés du PSL (*Process Specification Language*), qui est un framework pour décrire la structure des exécutions de processus, comme celui de [4]. Des ontologies pour les procédures peuvent être construites sur la base des spécifications ISO, comme présenté dans [5]. À l'exception du dernier, ces modèles sont complexes et seul le fragment d'entre eux qui traite du processus description est lié à notre étude.

Par conséquent, les concepts principaux suivants sont introduits dans notre ontologie pour décrire des procédures de maintenance. Ces dernières sont inspirées de la définition de la norme ISO et celles des travaux intérieurs :

Resource. Représente l'ensemble de ressources liées à la maintenance. Cette classe possède des sous-classes :

- *Component* : Désigne l'objet à maintenir, par exemple, une aile d'avion.
- *Device* et *Tool* : Désigne des appareils (par exemple, un régulateur de pression d'air) ou outils (par exemple, scieuse ou visseuse) utilisés pour la maintenance. Les informations sur outils utilisés pour la maintenance sont collectées à partir de différentes sources sur l'Internet, surtout les thésaurus. Chaque outil est considéré comme sous-classe de *Tool* et est enrichi à l'aide de DBPedia [4] (la recherche des entités est prise en charge par DBPedia Lookup [5]), une source dite LOD. De cette manière, chaque outil possède une description, est associé à plusieurs sujets (thème), et est lié à une ressource de DBPedia pour nous y diriger si nous avons besoin de plus d'information supplémentaire.
- *Aircraft* : Représente l'avion (par exemple, Airbus A380) ayant le composant à maintenir.

***Process*, *Task* et *Subtask*.** La classe *Process* représente les procédures de maintenance qui se composent de tâches (*Task*) successives. Une tâche peut contenir à son tour des sous-tâches (*Subtask*) et peut faire référence à une autre procédure.

2. Actuellement un modèle basé sur Spacy et la détection des *patternes* est en cours d'être entraîné
3. https ://www.aerospool.sk/downloads/RTC/AS-AMM-01-000_I1_R1_20180202.pdf
4. https ://dbpedia.org/
5. https ://github.com/dbpedia/dbpedia-lookup

Act. Identifie une activité qui a eu lieu, a lieu, ou devrait avoir lieu à l'avenir. Un *Act* peut-être associé à une tâche ou une sous-tâche. La liste des actions sont créé à l'aide du dictionnaire des verbes d'action introduite dans le paquet QDAP (*Quantitative Discourse Analysis Package* [6]). Chaque action (verbe) est considérée comme sous-classe de *Act* et possède éventuellement des actions équivalentes (synonymes), des extensions (hypernymes) ou des actions inverses (antonymes). Ces derniers, ainsi que sa définition, proviennent de WordNet [7]. Ces informations permettent de bien définir les actions et les relations entre elles en général ; et de les reconnaître dans de différents cas d'usage en particulier.

Context. Représente des informations supplémentaires sur une procédure ou une tâche. Il s'agit d'un avertissement, d'une condition préalable à l'action ou de la position précise du composant.

4 Représentation de la connaissances du domaine

Nous examinons ensuite comment représenter la connaissance du domaine, notamment l'information sur les composants d'avion afin de l'associer dans les procédures de maintenance et/ou demande de réparation. Dans le domaine aéronautique, à notre connaissance, il n'existe qu'une seule ontologie, appelée Aircraft Ontology [7], décrivant les composants et la structure d'avion. Cependant, l'ontologie ne couvre pas tous les composants d'avion dont nous avons besoin, comme les antennes dans notre cas d'étude ci-dessous. Ce seul travail n'est malheureusement pas adapté à la réutilisation. En effet, l'ontologie proposée ne contient que des composants de hauts niveaux, donc il n'existe pas des composants nécessaires. Nous avons choisi le système ATA [8] pour structurer les concepts. Le système est défini par Air Transport Association of America. Il permet de regrouper les systèmes aéronautiques dans des rubriques. Cette structuration permet de localiser le composant ainsi que d'identifier éventuellement des composants équivalents.

L'information sur les composants est récupérée sur le site aviationsourcingsolution [9]. Chaque composant concret récupéré (par exemple le composant dont le partN0 est 170-00834-803 est créé comme une classe dont le composant générique est *Main landing gear instl left*. Ce dernier est trouvé dans le chapitre *Landing Gear* (ATA 32) appartenant au groupe *Aircraft Systems*.

Il existe en outre de milliers de demandes de maintenances à analyser et à intégrer dans notre base de connaissances. Un modèle sera utilisé pour extraire des informations pertinentes comme les personnes ayant créé et approuvé la demande, la description des dommages ainsi que le composant concerné. Ainsi les concepts suivants sont proposés, en complément avec ceux ci-dessus :

Axiom	252.015
Logical axiom count	60.064
Declaration axioms count	60.046
Class count	60.034
Annotation Property count	12
Object property count	14
AnnotationAssertion	17.029

TABLE 1 – Quelques métriques de l'ontologie initiale

***Enterprise* et *Employee*.** Nous réutilisons l'ontologie Organisation [10]. Il s'agit d'une ontologie de base, recommandée par le W3C, pour les structures organisationnelles. Cette dernière, à son tour, réutilise le vocabulaire FOAF [11], une référence permettant de décrire des personnes et les relations qu'elles entretiennent entre elles.

***Damage* et *DamageType*.** Chaque demande de maintenance est créée pour maintenir ou réparer un composant qui a un certain dommage. Ce dernier appartient à un type de dommage. Nous avons dressé aussi une liste de types de dommage fréquemment rencontrés.

5 Résultat et exploitation

La Figure 2 décrit les concepts clés de notre ontologie [12]. Grâce à la méthodologie présentée, à l'heure actuelle, nous avons indexé plus de 54.000 composants d'avion, plus de 130 outils fréquemment utilisés et plus de 1.500 verbes d'action. La Table 1 liste quelques métriques de notre ontologie.

Application. À l'aide de la base de connaissances construites, plusieurs applications sont prévues :

- Génération automatique des illustrations graphiques : La connaissance sur les graphiques peut être rajoutée et associée aux éléments disponibles dans notre base. Un cas d'étude démontrant la faisabilité de cette proposition a été présenté dans [8]. Dans ce dernier, de nouveaux concepts ont été ajoutés pour représenter la connaissance sur les graphiques de façon qu'un graphique d'origine en format vectoriel 2D soit associé aux composants concernant une procédure de maintenance. Ce graphique d'origine permet de générer de nouveaux graphiques représentant différentes étapes de la procédure.
- Attribution automatique des tâches : Une autre application envisagée est d'utiliser des algorithmes d'apprentissage automatique pour proposer l'agent de maintenance le plus adapté en fonction de la nouvelle demande et l'historique des demandes dans le passé.
- Aide à la rédaction des manuels : Une autre piste à explorer est de proposer un outil aidant à la rédaction des documents de maintenance. Comme l'auto-

6. https ://www.rdocumentation.org/packages/qdap
7. https ://wordnet.princeton.edu/
8. https ://www.aerospaceunlimited.com/ata-chapters/
9. https ://www.aviationsourcingsolutions.com/
10. https ://www.w3.org/TR/vocab-org/
11. http ://xmlns.com/foaf/0.1/
12. Pour le moment, nous ne sommes pas en mesure de publier notre ontologie pour des raisons de confidentialité

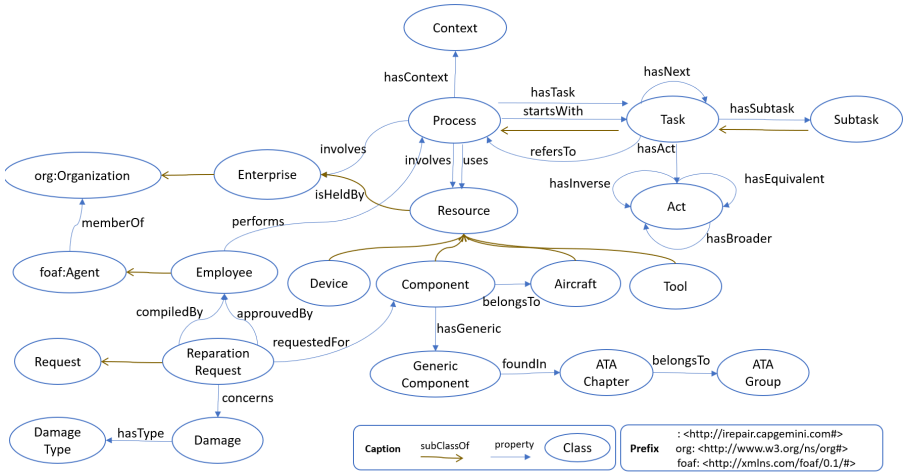


FIGURE 2 – Concepts clés de l'ontologie introduite dans IREPAIR

complétion, il proposera, de manière automatique, des compléments (actions, composants ou outils...) qui pourrait convenir au contexte et au texte rédigé.

6 Conclusion et perspectives

Nous avons présenté une méthodologie avec laquelle nous avons construit notre ontologie représentant la connaissance en maintenance aéronautique. L'ontologie développée évolue progressivement grâce à l'extraction des informations à partir des données non structurées par un modèle linguistique. Plusieurs cas d'étude sont prévus pour exploiter la base de connaissances.

D'un part, nous considérons à valider et étendre le modèle proposé pour prendre en compte plus d'information sur les composants. D'autre part, nous envisageons de compléter notre ontologie et la mettre en correspondance avec la classification disponible chez les fabricants tels qu'Airbus ou Boeing. Enfin, nous planifions d'étudier et d'appliquer des algorithmes plus sophistiqués et plus performants pour mettre en correspondance et avec précision les entités issues du TALN et les concepts disponibles dans notre base de connaissances.

Références

[1] Rospocher, M., Ghidini, C. & Serafini, L. An Ontology for the Business Process Modelling Notation.. *FOIS*. (2014)

[2] Karray, M., Chebel-Morello, B. & Zerhouni, N. A formal ontology for industrial maintenance. *Applied Ontology*. (2012)

[3] Chungoora, N., Young, R., Gunendran, G., Palmer, C., Usman, Z., Anjum, N., Cutting-Decelle, A., Harding, J. & Case, K. A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. *Computers In Industry*. (2013)

[4] Grüninger, M. Using the PSL ontology. *Handbook On Ontologies*. (2009)

[5] Fraga, A., Vegetti, M. & Leone, H. Semantic Interoperability among Industrial Product Data Standards using an Ontology Network.. *ICEIS (2)*. (2018)

[6] Annane, A., Aussenac-Gilles, N. & Kamel, M. BBO : BPMN 2.0 Based ontology for business process representation. *20th European Conference On Knowledge Management (ECKM 2019)*. (2019)

[7] Ast, M., Glas, M., Roehm, T. & Luftfahrt, V. Creating an ontology for aircraft design. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV. (2014)

[8] Hoang, T., Tran, B. & Mozafari, M. A Semantic Approach for Generating Graphical Representation from Aircraft Maintenance Text. *Proceedings Of The 14th International Joint Conference On Knowledge Discovery, Knowledge Engineering And Knowledge Management - Volume 3 : KMIS*. (2022)

[9] Maedche, A. & Staab, S. Ontology learning. *Handbook On Ontologies*. pp. 173-190 (2004)

[10] Gangemi, A. A Comparison of Knowledge Extraction Tools for the Semantic Web. *The Semantic Web : Semantics And Big Data*. pp. 351-366 (2013)

Développer des applications sémantiques en expliquant les conséquences de conception de la base de connaissances

Lortal Gaëlle[1]

[1] Thales Research and Technology, LRASC

gaelle.lortal@thalesgroup.com

Résumé

Les applications à base de sémantique intéressent de plus en plus les industriels, principalement parce qu'elles permettent le partage d'information et l'explicabilité des résultats. Afin de développer l'utilisation des applications sémantiques, nous proposons un outil de création de bases de connaissances (ontologies) pour l'industrie, c'est-à-dire pour non experts.

L'outil GLUON prend en compte (1) l'interaction nécessaire avec l'utilisateur (interaction linguistique grâce à un verbaliseur), (2) différentes logiques nécessaires à l'utilisateur (dans la mesure du possible) ainsi que (3) l'existant (création de base de connaissances de zéro, à base de texte, à base de réutilisation ou d'alignement/fusion). (4) La vérification des bases de connaissances créées se fait avant tout par conception, au fil de l'eau.

Mots-clés

Ontologie, Reasoners, Verbalizers

Abstract

Semantic-based applications are of increasing interest to industrialists, mainly because they allow the sharing of information and the explainability of results. To develop the use of semantic applications, we propose a tool to create knowledge bases (ontologies) for industry i.e. non-experts.

The tool GLUON takes into account (1) the necessary interaction with the user (linguistic interaction thanks to a verbalizer), (2) different logics necessary for the user (as much as possible) as well as (3) the existing (knowledge base creation from scratch, text-based, reuse-based or alignment/merge). (4) The verification of the created knowledge bases is done above all by design, as it happens.

Keywords

Ontology, Reasoners, Verbalizer

1 Introduction

Aujourd'hui, les entreprises comprennent les enjeux et les gains commerciaux liés aux produits et services à base de sémantique. Cependant, pour mettre en place des telles applications et services, il est nécessaire de développer le cœur de ces applications, la base de connaissances. Or, les connaissances sont propriétés des experts du domaine de l'application. Une des problématiques à résoudre est donc la création de base de connaissances par un expert domaine non expert des bases de connaissances.

Les outils support permettant aux experts de domaines de créer les bases de connaissances nécessaires aux applications et services à mettre en place doivent répondre à un certain nombre d'exigences fonctionnelles. L'outil doit prendre en compte (1) l'interaction nécessaire avec l'utilisateur (interaction linguistique grâce à un verbaliseur), (2) différentes logiques nécessaires à l'utilisateur (dans la mesure du possible) ainsi que (3) l'existant (création de base de connaissances de zéro, à base de texte, à base de réutilisation ou d'alignement/fusion). (4) La vérification des bases de connaissances créées doit se faire avant tout par conception, au fil de l'eau.

La partie 2 présente nos besoins. La partie 3 présente des travaux existants sur la création d'ontologie. Des méthodologies, outillées ou non, répondent à un principe sous-jacent qui implique en général le suivi d'étapes de création. La troisième partie présente l'outil développé et les premiers tests avant de conclure et de présenter les perspectives prenant en compte différents besoins opérationnels.

2 Besoins utilisateur

Nos utilisateurs travaillent dans le domaine de l'avionique. Ils sont experts, sans connaissance particulière des ontologies mais avec le plus souvent une forte connaissance en conception de systèmes (design authorities) ou en logiciel.

Ils ont donc besoin que leurs possibilités de conception ne soient pas entravées et que les outils proposés soient non seulement très flexibles mais s'intègrent à leurs activités.

Il s'agit aussi de ne pas les écraser avec l'acquisition rapide de nouvelles connaissances tant sur les ontologies et les raisonnements que sur les applications sémantiques en général. Les interactions Homme-Machine sont donc pensées avec soin pour une utilisation naturelle.

La fonctionnalité principale doit rapprocher les utilisateurs finaux de la construction de leur ontologie en leur expliquant les éléments qu'ils sont en train de représenter en langue naturelle. Toutefois, pour ce faire, les experts ont malgré tout besoin de fonctionnalités de bas niveaux de construction d'ontologies.

Pour résoudre cette problématique, nous avons développé un module nommé GLUON qui est l'acronyme de « Génération de Liens Unifiant une Ontologie » dans le but de rendre les ontologies utilisables par des non experts.

Comme notre objectif est de simplifier l'approche, nous avons

proposé en premier lieu le format d'ontologie le plus utilisé, à savoir OWL 2 DL. Le pont que nous construisons se situe entre un format logique et une interaction simple et naturelle.

3 Travaux antérieurs sur la création d'ontologie

Depuis les années 1990 [6], l'ontologie et sa création est scrutée, déconstruite pour être mieux reconstruite ou réutilisée. De Noy et Hafner en 1997 [8] à aujourd'hui, un grand nombre d'état de l'art sur la construction d'ontologies a vu le jour.

Les sources utilisées pour la construction des ressources terminologiques peuvent être structurées (base de données), semi-structurées (dictionnaire, corpus annoté) ou brutes (texte). De même, la constitution peut être semi-automatique ou guidée par des principes (manuelle) selon le but dans lequel l'ontologie s'inscrit, respectivement, son intégration dans un système transparent, ou sa construction elle-même comme objectif. Les méthodologies, outillées ou non, utilisées dans ces constructions répondent à un principe sous-jacent qui implique le suivi d'étapes de création. Le cycle classique de développement d'une ontologie est constitué des étapes de spécification, planification, conceptualisation, formalisation, implémentation. Cependant, des travaux récents font un retour d'expérience sur le mise en place de ces méthodologies. Dans [9] les auteurs concluent que le développement de la structure et des instances ne se produit pas séparément mais conjointement et qu'au final un cycle de développement stéréotypé ne peut être clairement identifié.

Ainsi, les méthodologies existantes, même déclinées en modules prenant en compte un cycle de construction itératif voire des étapes facultatives montrent leurs limitations quand il s'agit de prendre en compte les activités industrielles. Il est indispensable dans notre cas que la méthodologie considère un cycle de vie de l'ontologie de domaine dans la vie du produit ou service, et non seulement un cycle de construction. Cela sera fait dans un deuxième temps avec la construction d'un système complet encapsulé dans les plateformes métier existantes.

Le fait qu'un cycle de développement stéréotypé ne peut être clairement identifié est problématique et nous indique que pour respecter la liberté de conception des experts, il s'agit d'avoir une grande flexibilité dont la seule limite sera la vérification de la base de connaissances créée, de préférence par conception et au fil de l'eau, pour donner une souplesse à l'activité de l'expert. L'outil se définit donc par des fonctionnalités basiques de création des éléments constitutifs d'une ontologie, alliées à une vérification au fil de l'eau de sa constitution permise par l'utilisation de raisonneurs (un programme qui effectue un raisonnement sur une ontologie ou une base de connaissances). Le raisonneur s'appuie sur les connaissances ainsi que sur l'ensemble des règles de l'ontologie pour inférer (déduire logiquement) de nouvelles connaissances ajoutées au contenu de la base de connaissances.

Pour assurer la qualité de l'ontologie, il est nécessaire de gérer les incohérences et les incertitudes dans les ontologies dans les applications du monde réel [1]. Une ontologie incohérente signifie qu'une erreur ou un conflit existe dans une ontologie, et que certains concepts ne peuvent être interprétés correctement. La cohérence assure des liens ou des relations entre des concepts qui ont un sens explicite [3]. À titre d'exemple, on considère que deux classes déclarées disjointes n'héritent pas l'une de l'autre : une incohérence sera donc clairement levée si une instance doit hériter des deux classes disjointes à la fois. Par exemple, en avionique, un pilote effectuant un vol effectue sa « Mission ». Si Mission et Insatisfiable Mission sont deux classes disjointes et qu'une instance m:Mission est une instance d'un m':UnsatisfiableMission) alors la représentation est inadmissible logiquement et lèvera une erreur.

Après une étude détaillée des raisonneurs pour les ontologies [7], et à partir de nos différents critères pour trouver un bon compromis entre rapidité, expressivité et précision, Pellet, basé sur l'algorithme Tableau et supportant OWL2 DL est notre meilleur candidat. Néanmoins, pour les tests unitaires, nous utiliserons également ELK basé sur l'algorithme consequence-based, supportant un profil moins expressif de OWL, OWL2 EL, mais effectuant des traitements en un temps record, pour des tests plus légers. Une fois défini un raisonneur pour "comprendre" et lever les incohérences, il s'agit de l'inclure d'une façon transparente à notre outil afin que les utilisateurs puissent comprendre aussi sans être dépassés.

Afin d'offrir une solution viable entre la cohérence axiomatique et l'interaction avec un utilisateur, nous proposons d'utiliser un verbaliseur [7]. En effet, pour faciliter la compréhension des ontologies, plusieurs verbaliseurs d'ontologies ont été développés [11]. Les verbaliseurs traduisent généralement les axiomes de l'ontologie un par un dans un langage contrôlé malheureusement sans prêter attention à la cohérence du résultat en tant que texte.

ACE [4] est bien adapté à la verbalisation d'ontologies OWL, car les axiomes sont exprimés en phrases compactes n'explicitant que des phrases compatibles avec l'expressivité de OWL. OWL Verbalizer basé sur ACE permet une intégration dans notre plateforme. Une comparaison de systèmes de verbalisation OWL est disponible dans [7].

4 Spécifications de l'outil

Cette section traite de la solution proposée pour GLUON (Génération de Liens Unifiant une Ontologie) et la figure ci-après montre l'architecture de haut niveau de GLUON.

La nouveauté de la solution réside dans la l'intégration d'un verbaliseur OWL et dans le fait que des phrases explicatives sont générées pour décrire l'ontologie. GLUON est constitué d'un constructeur d'ontologie, d'un système de raisonnement et d'un générateur de langue naturelle (verbaliseur).

4.1 Constructeur d'ontologie

Ce composant permet de créer une ontologie à partir de rien (from scratch) ou de charger une ontologie déjà conçue via le gestionnaire d'ontologie, qui est un groupe d'instructions à partir d'un moteur d'ontologie. Ce bloc permet la création de concepts, de relations et d'individus à ajouter à l'ontologie en question, voire d'appliquer Opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) sur le contenu de l'ontologie pour assurer la persistance des données ontologiques. L'utilisateur a également la possibilité d'ajouter des règles (ex. règles SWRL) à son ontologie en les adaptant au domaine et à la

logique métier qui organisent son domaine d'expertise. Concernant le moteur d'ontologie, nous avons opté pour l'API OWL comme bibliothèque de manipulation d'ontologies. Ce module est présent dans de nombreux outils et une grande communauté l'utilise, de sorte que la documentation et les références sont disponibles.

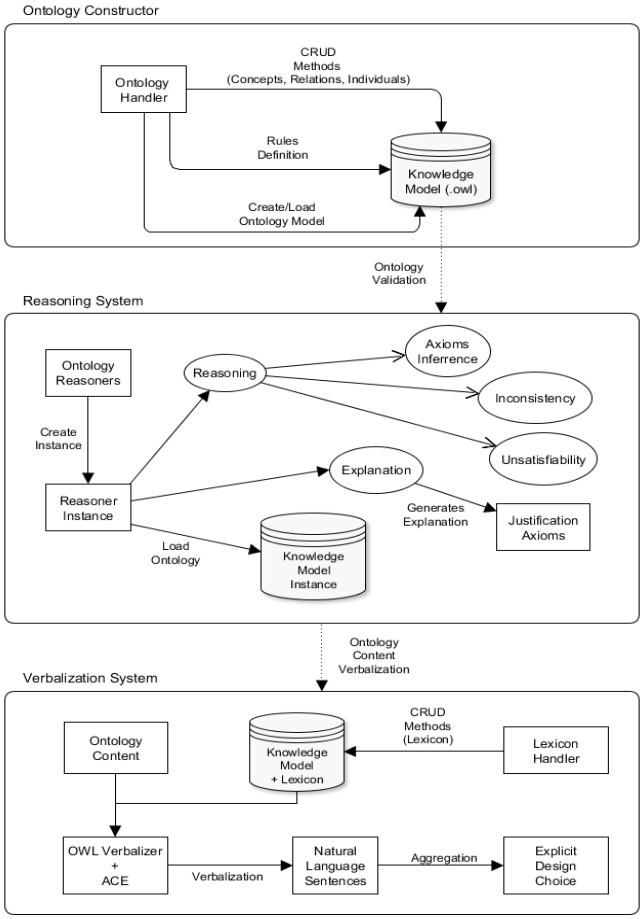


Figure 1 : Architecture de haut niveau de GLUON

4.2 Système de raisonnement

Ce composant est un ensemble de sous-systèmes permettant l'utilisation d'un raisonneur autonome ou de plusieurs raisonnements parallèles. En plus de raisonner sur une ontologie (inférences, détection des incohérences et insatisfiabilités), le raisonneur permet d'extraire les axiomes en OWL contenus dans l'ontologie. Ce module vérifie la cohérence de l'ontologie après chaque opération ou changement que l'utilisateur applique à l'ontologie en cours de conception. Le module génère des explications sous la forme d'axiome OWL ou d'une liste d'axiomes OWL. Cette explication est générée après chaque opération appliquée sur l'ontologie courante en utilisant le raisonneur définit par défaut dans un premier temps Openllet et à terme possibilité d'utiliser plusieurs raisonneurs, par exemple. Pellet, ELK, et bientôt DeLorean [2], DRAON [5] et STARE[1]...) pour la génération d'explications. Ces raisonneurs ont été choisis pour leur application à différents formats d'ontologie, leur utilisation de différents algorithmes de raisonnement ou encore leurs raisonnements sur des logiques différentes. Pour le moment, Pellet (Openllet) basé sur l'algorithme Tableau prend en charge OWL2 DL et ELK basé sur l'algorithme Consequence-based prend en charge OWL2 EL. Ce choix nous permet de couvrir dans un premier temps les expressivités DL et EL et permet également la justification, la prise en charge des règles SWRL, le raisonnement au niveau de la Abox et l'utilisation de l'API OWL dans le cadre de processus de raisonnement.

4.3 Système de verbalisation

Le système de verbalisation fait partie intégrante du système. Les phrases sont générées à partir de ce module avec le contenu de l'ontologie. Pour ce projet, nous avons utilisé OWL Verbalizer comme serveur HTTP et en même temps exploité les méthodes Java du module pour la verbalisation hybride. Les lexiques peuvent soit être générés à partir du noms associés aux concepts, relations et individus (par exemple #Aircraft, #flownWith, #MissionA, etc.) ou être définis par l'utilisateur via le module gestionnaire de lexique en l'adaptant aux différentes formes qu'il peut prendre (singulier, pluriel, passé forme participe pour les verbes, etc.). Noms, adjectifs et verbes peuvent être définis pour une meilleure génération ou importés en ressource prédéfinie grâce aux méthodes de gestion de lexique. Ainsi, OWL Verbalizer permet aux experts sans connaissance en logique de lire et comprendre les ontologies. Évidemment certains inconvénients subsistent, principalement la difficulté à agréger les phrases dans le cas de la verbalisation d'une liste d'axiomes.

5 Expérimentation de l'outil

5.1 Cas d'utilisation

Nous avons évalué l'utilisabilité et la performance des différents choix faits de GLUON sur la base de scénarios de validation proposés par des experts de l'Air Traffic Management. Ceux sont 3 cas d'utilisation (UC) d'une mission aérienne et modélisés sous forme d'ontologie :

- Défaillance du système
- Déviation en raison de la fermeture de la piste d'arrivée
- Événement météorologique

Chaque scénario reflète un incident, qui peut survenir pendant une mission aérienne. Voici la transcription ontologique du 1er scénario seulement pour exemple :

System failure (MissionSMAScenarioA)

Property assertions:

```
Individual: MissionSMAScenarioA
MissionSMAScenarioA Type Mission
MissionSMAScenarioA hasActiveAirportDeparture LFPB
MissionSMAScenarioA hasActiveAirportArrival KTEB
MissionSMAScenarioA hasActiveArrivalRunway KTEB06
MissionSMAScenarioA isFlownWith F-DEV1
F-DEV1 hasFuel FDEV1CurrentFuel
FDEV1CurrentFuel Type Fuel
FDEV1CurrentFuel hasFuelValue "0.0"
Mission DisjointWith UnsatisfiableMission
LowFuel SWRL Rule:
```

[1] https://gitlab.inria.fr/DLreasoners/stare

```
Mission(?m) ^ Fuel(?fcv) ^ IsFlownWith(?m, ?a)
^hasFuel(?a, ?fcv) ^ hasFuelValue(?fcv, 0.0) ->
hasUrgency(?m, "3")
Urgency SWRL Rule:
Mission(?m) ^ hasUrgency(?m, ?urg) ->
UnsatisfiableMission(?m)
Explanation: F-DEV1 has fuel with fuel value 0.0
Implies: MissionSMAScenarioA hasUrgency "3" ->
MissionSMAScenarioA Type UnsatisfiableMission
```

5.2 Résultats

GLUON permet des sorties en Langage Contrôlé des axiomes OWL présentés ci-dessus. Notre plan d'évaluation est en deux volets. La première partie est le test unitaire sur chacun des scénarios ci-avant sur les besoins des experts en avionique. Nous présentons ici le scénario Alpha c'est-à-dire l'ontologie consistante puis le scénario révélant des échecs. Une évaluation sera à conduire dans une deuxième étape sur une nouvelle mission (Sick Passenger Onboard) par les experts en avionique sur une ontologie qu'ils auront créée avec GLUON.

Scénario Alpha : (Consistent ontology verbalization)

```
F-DEV1 is an Aircraft.
F-DEV1 has fuel FDEV1 current fuel.
FDEV1 current fuel is a fuel.
FDEV1 current fuel have fuel value 0.0.
KTEB is an airport.
KTEB06 is an operational runway.
LFPB is an airport.
Mission SMA A is a mission.
Mission SMA A has active airport arrival KTEB.
Mission SMA A has active airport departure LFPB.
Mission SMA A has active arrival runway KTEB06.
Mission SMA A is flown with F-DEV1.
```

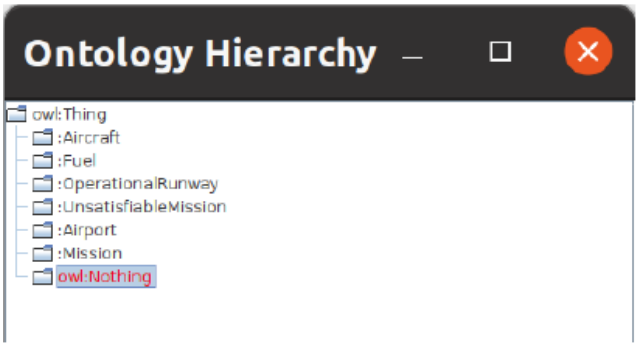


Figure 2: Hiérarchie de l'ontologie utilisée

Explication générée par GLUON pour le scenario « System failure inconsistency »:

```
You know that Mission SMA A is flown with F-DEV1 and FDEV1
current fuel has fuel value 0.0 and Mission SMA A is a
mission and No mission is an unsatisfiable mission and F-
DEV1 has fuel FDEV1 current fuel and FDEV1 current fuel
is a fuel.
```

Les textes générés correspondent bien aux incohérences que les experts souhaitaient lever dans leurs 3 scénarios de missions aériennes. Malgré la simplicité de la plupart des axiomes dans notre ontologie, leur verbalisation est complexe. Certains axiomes complexes ne pourront être verbalisés par OWL Verbalizer.

Malgré tout, la présentation reste plus lisible que la syntaxe standard des logiques DLs.

6 Conclusion et perspectives

Permettant le partage d'information et l'explicabilité des résultats, les applications sémantiques sont un facteur clés de l'argumentation commerciale aujourd'hui. Afin de développer leur utilisation, GLUON permet aux experts industriels de concevoir et modéliser l'application proposée.

Pour cela, GLUON facilite l'interaction utilisateur/formalisme logique via un verbaliseur et à venir il proposera des modules de construction d'ontologies à base de texte et de données structurés (alignement d'ontologie, extension d'ontologies de haut-niveau). La problématique de vérification des ontologies est prise en compte en natif par la proposition d'explications des choix de conception de l'utilisateur mais des travaux méthodologiques sont en cours sur la validation et la vérification des bases de connaissances et des raisonnements.

Une première évaluation positive nous pousse à mettre en place un approfondissement des perspectives sur les axes raisonnement et logique et sur l'ajout de module d'alignement.

7 Références

[1] Abburu, S. (2012). A survey on ontology reasoners and comparison. International Journal of Computer Applications, 57:33–39

[2] Bobillo, F., Delgado, M., & Gómez-Romero, J. (2008). DeLorean: A Reasoner for Fuzzy OWL 1.1. In URSW.

[3] Euzenat, J., & Shvaiko, P. (2007). Ontology matching (Vol. 18). Heidelberg: Springer

[4] Kaljurand, K. (2007). Attempto controlled english as a semantic web language. University of Tartu

[5] Le Duc, C., Lamolle, M., Zimmermann, A., & Curé, O. (2013). DRAOn: A Distributed Reasoner for Aligned Ontologies. In ORE (pp. 81-86)

[6] Mars, NJI(Ed.), 1994. Workshop comparison of implemented ontologies, In ECAI, 8–12 08, Amsterdam, NL

[7] Mejdoul, Z. and Lortal, G. (2022). GLUON: A Reasoning-based and Natural Language Generation-based System to Explicit Ontology Design Choices. In IJC3K ISBN 978-989-758-614-9; SciTePress, pages 228-236.

[8] Noy, N. F., & Hafner, C. D. (1997). The State of the Art in Ontology Design: A Survey and Comparative Review. AI Magazine, 18(3), 53. https://doi.org/10.1609/aimag.v18i3.1306

[9] Reiz, A. and Sandkuhl, K. (2022). Debunking the Stereotypical Ontology Development Process. In IJC3K ISBN 978-989-758-614-9; SciTePress, pages 82-91

[10] Schwitter, R. (2010). Controlled natural languages for knowledge representation. In Coling 2010: Posters (pp. 1113-1121).

Système de recommandations basé sur les contraintes pour les simulations de gestion de crise

Ngoc Luyen Le[1,2], Jinfeng Zhong[3], Elsa Negre[3], Marie-Hélène Abel[1]

[1] Université de technologie de Compiègne, CNRS, Heudiasyc
(Heuristics and Diagnosis of Complex Systems), CS 60319 - 60203 Compiègne Cedex, France.
[2] Vivocaz, 8 B Rue de la Gare, 02200, Mercin-et-Vaux, France.
[3] Université Paris Dauphine-PSL, Université Paris Sciences et Lettres,
CNRS UMR 7243, LAMSADE, Paris, France.

Résumé

Dans le cadre de l'évacuation des populations, certains citoyens/bénévoles peuvent et souhaitent participer à l'évacuation des populations en difficulté en venant prêter main-forte aux véhicules d'urgence/évacuation avec leurs propres véhicules. Une manière de cadrer ces élans de solidarité serait de pouvoir répertorier en temps réel les citoyens/bénévoles disponibles avec leurs véhicules (terrestres, maritimes, aériennes, etc.), de pouvoir les géolocaliser en fonction des zones à risques à évacuer et de les ajouter aux véhicules d'évacuation. Parce qu'il est difficile de proposer un système opérationnel temps réel efficace sur le terrain en situation de crise réelle, nous proposons dans ce travail d'ajouter un module de recommandation de couples conducteur/véhicule (avec leurs spécificités) à un système de simulation de gestion de crise. Pour ce faire, nous avons choisi de modéliser et de développer un système de recommandations basé sur des contraintes s'appuyant sur une ontologie pour les simulations de gestion de crise.

Mots-clés

Graphe de connaissances, Système de recommandations, Ontologie, Simulations, Gestion de crise

Abstract

In the context of the evacuation of populations, some citizens/volunteers may want and be able to participate in the evacuation of populations in difficulty by coming to lend a hand to emergency/evacuation vehicles with their own vehicles. One way of framing these impulses of solidarity would be to be able to list in real-time the citizens/volunteers available with their vehicles (land, sea, air, etc.), to be able to geolocate them according to the risk areas to be evacuated, and adding them to the evacuation/rescue vehicles. Because it is difficult to propose an effective real-time operational system on the field in a real crisis situation, in this work, we propose to add a module for recommending driver/vehicle pairs (with their specificities) to a system of crisis management simulation. To do that, we chose to model and develop an ontology-supported constraint-based recommender system for crisis management simulations.

Keywords

Knowledge graph, Constraint-based Recommender System, Ontology, Simulation, Crisis management

1 Introduction

Dans le contexte de l'évacuation des populations, les ressources publiques traditionnelles telles que les ambulances et les hélicoptères de gendarmerie (avec des conducteurs professionnels) peuvent être limitées et mal positionnées pour atteindre toutes les personnes dans le besoin. Dans de telles situations, il est nécessaire d'explorer des ressources d'évacuation alternatives. Les ressources citoyennes, en revanche, sont généralement plus dispersées et donc plus accessibles. De plus, de nombreux citoyens/bénévoles [1] peuvent être disposés à aider à l'évacuation en utilisant leurs propres véhicules. Par exemple, un propriétaire de minibus avec une capacité de 9 passagers pourrait potentiellement évacuer 8 personnes supplémentaires, augmentant considérablement la capacité du processus d'évacuation. De même, un propriétaire de bateau avec une capacité de 6 passagers pourrait aider à évacuer 5 personnes en cas d'inondation. Malheureusement, les recherches existantes sur la gestion de crise (par exemple, la simulation d'évacuation) [3,5,7,9,11,12] se sont principalement concentrées sur l'utilisation de ressources publiques telles que les ambulances et les camions de pompiers. Cependant, dans certains cas, ces ressources peuvent ne pas être disponibles en raison d'une demande élevée ou de la localisation éloignée de la zone touchée. De plus, la localisation des ressources publiques peut également être impactée par une crise, aggravant la pénurie de ressources.

En réalisant ce travail, nous apportons les contributions suivantes : (i) nous étudions le développement d'une ontologie pour aider à organiser des vocabulaires partagés, standardiser les connaissances liées à la gestion de crise et faciliter la mise en œuvre d'un système de recommandations basé sur des contraintes. En utilisant cette ontologie, nous pou-

1. Dans cet article, nous nous concentrons sur les conducteurs citoyens/bénévoles (parmi les citoyens qui ne sont pas impliqués dans la gestion de crise), ainsi que sur leurs propres véhicules, que nous résumerons par le terme de "conducteurs/véhicules citoyens/bénévoles".

vons rationaliser le processus de réutilisation des informations afin d'améliorer l'efficacité du système de recommandation basé sur des contraintes ; (ii) nous formulons le problème de distribution de véhicules lors de la mise à l'abri des populations comme un problème de recommandation, ce qui nous permet d'incorporer différentes techniques de recommandation pour allouer efficacement les ressources. Plus précisément, notre système de simulation de crise basé sur une ontologie pour la mise à l'abri des populations vise à consolider les ressources citoyennes pour aider à mettre les populations à l'abri lors d'une crise, en particulier dans les situations où les ressources publiques sont insuffisantes. Le reste de cet article est organisé comme suit. La section suivante présente la construction de l'ontologie et notre modèle de système de recommandations basé sur des contraintes pour les simulations de gestion de crise. Dans la troisième section, nous présentons notre prototype et son application sur un cas d'utilisation détaillé. Enfin, nous concluons en proposant quelques pistes de travaux futurs.

2 Notre approche

2.1 Formulation du problème

Nos travaux visent à modéliser et proposer un système de gestion des ressources conducteur/véhicule pour l'évacuation des populations touchées par une crise. Deux problèmes clés sont abordés : ($P1$) l'organisation des données et informations relatives aux ressources conducteur/véhicule, et ($P2$) la recommandation de solutions optimales en tenant compte des contraintes de capacité, de temps de réponse et de contexte. Le problème ($P1$) se concentre sur le choix d'un modèle approprié pour organiser les données et l'information dans le contexte de la gestion de crise. La modélisation ontologique est utilisée pour capturer et représenter les concepts et les relations du domaine de la gestion de crise. Le problème ($P2$) concerne la conception et le développement d'un système de recommandations capable de proposer des solutions d'allocation de ressources adaptées à chaque situation. Les technologies de recommandation basées sur les connaissances et l'ontologie permettent de prendre en compte les exigences spécifiques des points de secours [1] et de calculer des recommandations pertinentes en utilisant les connaissances sur le contexte et les ressources disponibles.

Definition 1 *Un système de recommandations basé sur des contraintes pour l'allocation des ressources est défini en utilisant 4 ensembles : l'ensemble des ressources mobiles*[2] $\mathcal{R}$ *avec leurs caractéristiques/attributs, l'ensemble des points de secours et leurs besoins* $\mathcal{P}$*, l'ensemble des abris* $\mathcal{S}$ *avec leurs caractéristiques/attributs, et l'ensemble des contraintes* $\mathcal{C}$*. Une recommandation de solution pertinente est calculée en fonction de l'élément concret, des ensembles* $\mathcal{R}$*,* $\mathcal{P}$*, et* $\mathcal{S}$ *de sorte que les contraintes spécifiées* $\mathcal{C}$ *soient satisfaites*

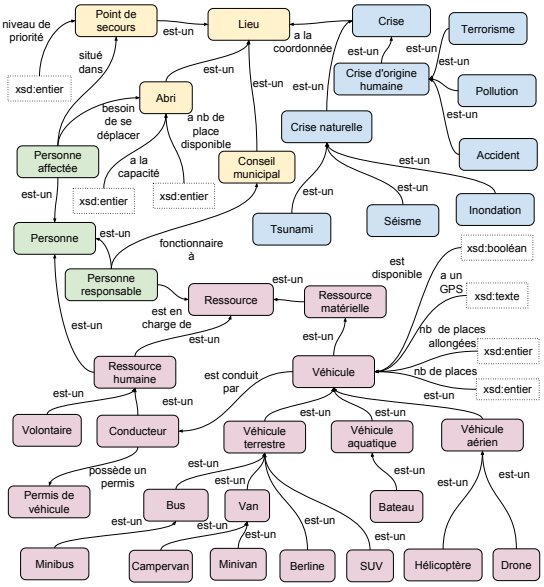


FIGURE 1 – Un extrait de l'ontologie concernant les ressources conducteur/véhicule, les lieux et les personnes concernées dans la gestion d'une crise.

Definition 2 *Une tâche de recommandation pour l'allocation des ressources mobiles est définie comme un problème de satisfaction de contraintes* $(\mathcal{R}, \mathcal{P}, \mathcal{S}, \mathcal{C})$ *basé sur l'attribution et le calcul du nombre de ressources mobiles dans l'ensemble* $\mathcal{R}$ *allouées à un point de secours dans* $\mathcal{P}$ *de sorte qu'il satisfait et ne viole aucune des contraintes dans* $\mathcal{C}$*.*

Une recommandation de solution optimale pour l'allocation des ressources mobiles dans $\mathcal{R}$ proposera une liste de ressources mobiles disponibles utilisées pour transférer les évacués des points de secours $\mathcal{P}$ vers les abris $\mathcal{S}$ avec un temps de déplacement minimum. Dans la prochaine section, nous détaillerons le développement d'une ontologie en gestion de crise pour les ressources et les facteurs liés.

2.2 Construction de l'ontologie

Dans notre étude, nous avons utilisé la Méthodologie Agile pour le Développement d'Ontologie (AMOD) [1], qui comprend trois phases distinctes : préliminaire, développement et post-développement, chacune contribuant à la réalisation progressive de l'ontologie. Dans la phase préliminaire, l'objectif principal de la construction de l'ontologie est de fournir un modèle standard avec une terminologie et un vocabulaire pour collecter des informations sur les ressources disponibles (par exemple, les véhicules, les conducteurs) qu'une organisation (par exemple, le Conseil Municipal) mobilisera pour évacuer les personnes affectées lors d'une crise. Les concepts centraux de l'ontologie sont définis sur la base des entités importantes avec leurs caractéristiques dans le contexte de la gestion de crise.
Pendant la phase de développement, nous organisons des sprints [3] et choisissons de développer notre ontologie en

1. Un point de secours est un lieu spécifiquement désigné, au sein d'une situation de crise, où les personnes peuvent se rendre pour obtenir de l'aide, des soins médicaux ou être évacuées.

2. Les ressources mobiles se limitent aux ressources citoyennes dans le cadre de notre travail.

3. Un sprint est une unité de temps fixe pendant laquelle un ensemble spécifique de tâches doit être accompli.

s'appuyant sur l'ontologie *ISyCri* [2] en utilisant des concepts liés à la description de la crise, des personnes touchées et des ressources. Plus précisément, comme illustré par la Figure 1, nous adaptons et développons notre modèle ontologique autour de trois principales entités : les ressources, les personnes et les lieux. Premièrement, les ressources sont distinguées selon qu'elles sont humaines, matérielles ou mobiles. Dans notre cas, les ressources humaines sont les citoyens/bénévoles qui participent aux opérations de secours et d'évacuation. Tandis que les ressources matérielles incluent les catégories de véhicules et leurs informations descriptives. Une ressource mobile sera représentée par une association par défaut d'un véhicule et d'un conducteur (c'est-à-dire une paire de ressources conducteur/véhicule). Deuxièmement, l'identification et l'organisation des lieux joue un rôle extrêmement important dans notre cas. Chaque lieu doit être spécifiquement identifié avec des informations de localisation. En général, les lieux sont séparés en points de secours et abris. Les points de secours sont des sites où les personnes affectées se regroupent et sont acheminées vers un abri par une ressource mobile.

Enfin, les populations peuvent être distinguées entre les populations affectées et les ressources humaines. Les populations affectées sont les populations vulnérables lors de la crise, et elles ont besoin d'être déplacées vers un abri. Tandis que les ressources humaines peuvent être des conducteurs qui utilisent leur véhicule pour participer aux activités d'évacuation. En général, la représentation d'une personne par l'ontologie est utile pour rassembler les ressources humaines et les informations sur les personnes affectées dans les étapes préalables et postérieures à la crise.

2.3 Système de recommandation

Pendant la mise à l'abri des populations, la distribution efficace des conducteurs/véhicules citoyens/bénévoles disponibles vers les zones touchées est un sujet de recherche crucial. Ce problème peut être considéré comme un problème de recommandation, où les conducteurs/véhicules citoyens/bénévoles sont traités comme des éléments à recommander et les points de secours sont traitées comme des utilisateurs auxquels les conducteurs/véhicules citoyens/-bénévoles doivent être recommandés. Dans cette section, nous fournissons une description détaillée du système de recommandations basé sur les contraintes pour les simulations de gestion de crise. Les systèmes de recommandations basés sur les contraintes génèrent des recommandations en identifiant les éléments qui répondent à un ensemble de contraintes explicites prédéfinies. Dans notre cas, le système de recommandations vise à générer des paires de conducteurs/véhicules pour chaque point de secours, en veillant à ce que les conducteurs/véhicules affectés à chaque point de secours aient une capacité suffisante pour évacuer la population tout en minimisant le temps nécessaire pour atteindre les points de secours (pour plus de détails [10]).

Lorsque plusieurs solutions sont disponibles, notre algorithme renvoie celle qui utilise moins de véhicules pour réduire le temps total nécessaire et le risque d'embouteillages. Nous calculons T_{CV-RP} avec *OSMNX* [4], un package Python qui permet de télécharger des données géospatiales depuis *OpenStreetMap* : le système énumère exhaustivement toutes les solutions possibles et sélectionne celle qui utilise le nombre minimal de véhicules pour soulager les embouteillages. Pour réduire le temps de calcul requis pour générer la liste de recommandations, nous pré-calculons le temps estimé entre chaque point. Nous avons essayé *Google Maps API*, et il s'est avéré que son utilisation prenait plus de temps que OpenStreetMap pour calculer le temps estimé entre deux points. Nous avons donc adopté ce dernier. Plus précisément, nous avons utilisé les outils de recherche opérationnelle (OR-Tools) [8] pour l'optimisation combinatoire, conçus pour trouver la solution optimale à un problème à partir d'un ensemble extrêmement large de solutions possibles. Il convient de noter que les ressources publiques peuvent être considérées comme un cas particulier dans notre contexte. Les véhicules publiques sont généralement garés à des points fixes, tandis que les conducteurs/-véhicules sont souvent dispersés. Par conséquent, les véhicules publiques tels que les ambulances peuvent également être intégrés à notre contexte.

3 Prototype et cas d'étude

Suivant les directives de recherche en science de la conception proposées par [6] : La recherche en science de la conception doit produire un artefact viable sous la forme d'un construit, d'un modèle, d'une méthode, ou d'une instanciation. Dans cette section, nous présentons un prototype d'un système qui aide à recommander des ressources conducteur/véhicule pendant une crise. Nous présenterons d'abord l'architecture du système qui met en œuvre le système de recommandations basé sur des contraintes, comment le système fonctionne, puis nous décrivons un scénario pour illustrer l'utilité de notre système.

Comme illustré par la Figure 2, notre système est basé sur une architecture à quatre couches. La couche d'interaction comprend une interface mobile qui recueille la disponibilité des conducteurs citoyens/bénévoles pendant une crise, et une interface web permettant aux décideurs d'interagir avec le système et de préciser les informations de chaque point de secours. La couche d'intelligence, le cœur du système, calcule la liste des recommandations satisfaisant certaines contraintes et les affiche aux décideurs. La couche de service calcule les coordonnées de chaque point de secours à partir de sa localisation géographique et estime le temps nécessaire pour que chaque paire conducteur/véhicule atteigne le point de secours. Enfin, la couche de données contient une base de connaissances soutenue par une ontologie pour le domaine de la gestion de crise, modélisant et stockant toutes les informations et données nécessaires. A titre d'illustration, prenons une crise d'inondation qui se produit dans la ville de Compiègne et qui nécessite une évacuation rapide des personnes vulnérables vers des abris. Imaginons que le conseil municipal dispose (1) d'une liste de conducteurs citoyens/bénévoles associés et de leur

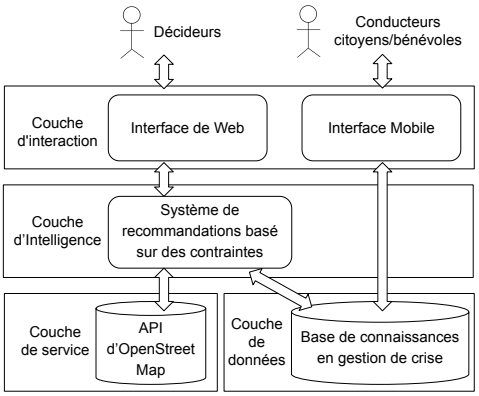


FIGURE 2 – L'architecture du système

véhicule personnel pouvant être sollicités en situation d'urgence ; (2) d'abris d'accueil d'urgence. A l'aide d'une interface Web, les décideurs devant gérer l'évacuation fournissent au système les informations requises pour chaque point de secours, notamment le nombre de personnes et de personnes handicapées à évacuer, ainsi que leur niveau de priorité. Le système consulte alors sa base de connaissances afin d'identifier les ressources conducteurs/véhicules disponibles. En utilisant les données d'OpenStreetMap, il calcule les temps de déplacement estimés entre les véhicules et les points de secours, en cherchant à les minimiser. Enfin, le système recommande aux gestionnaires de l'évacuation la liste optimale des paires conducteur/véhicule afin de les aider dans leur prise de décision. Cette approche permet d'optimiser l'allocation des ressources pour évacuer les personnes vulnérables dans les délais les plus courts vers les abris.

4 Conclusion et perspectives

Cet article présente un système de recommandations destiné à aider les décideurs à allouer des paires de conducteur/véhicule citoyens/bénévoles, lorsque les ressources publiques sont insuffisantes. Le système, structuré en quatre couches modulaires interconnectées, utilise une ontologie pour la structuration et le stockage des données, applique OpenStreetMap pour le calcul du temps et de la distance entre deux points géographiques, génère des recommandations pour chaque point de secours et facilite les interactions entre les décideurs et le système de recommandations. Pour l'avenir, nous envisageons d'enrichir l'ontologie pour mieux gérer les crises, d'ajouter davantage de contraintes pour une modélisation plus réaliste de la crise, de construire un système dynamique pour la réutilisation des ressources en temps réel et d'intégrer notre système dans une simulation basée sur des agents pour en évaluer les aspects sociotechniques dans différents scénarios de gestion de crise.

Remerciements

Cette recherche a été financée par l'Agence National de la Recherche (ANR) et par l'entreprise Vivocaz au titre du projet France Relance – préservation de l'emploi R&D (ANR-21-PRRD-0072-01).

Références

[1] Abdelghany Salah Abdelghany, Nagy Ramadan Darwish, and Hesham Ahmed Hefni. An agile methodology for ontology development. *International Journal of Intelligent Engineering and Systems*, 12(2) :170–181, 2019.

[2] Frédérick Bénaben, Chihab Hanachi, Matthieu Lauras, Pierre Couget, and Vincent Chapurlat. A metamodel and its ontology to guide crisis characterization and its collaborative management. In *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Washington, DC, USA, May*, 2008.

[3] Gary Bennett, Lili Yang, and Boyka Simeonova. A heuristic approach to flood evacuation planning. 2017.

[4] Geoff Boeing. Osmnx : A python package to work with graph-theoretic openstreetmap street networks. *Journal of Open Source Software*, 2(12), 2017.

[5] Ahmed T Elsergany, Amy L Griffin, Paul Tranter, and Sameer Alam. Development of a geographic information system for riverine flood disaster evacuation in canberra, australia : Trip generation and distribution modelling. In *ISCRAM*, 2015.

[6] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.

[7] Sarika Jain, Sonia Mehla, and Apoorv Gaurav Agarwal. An ontology based earthquake recommendation system. In *International Conference on Advanced Informatics for Computing Research*, pages 331–340. Springer, 2018.

[8] Serge Kruk. *Practical Python AI Projects Mathematical Models of Optimization Problems with Google OR-Tools*. Springer, 2018.

[9] Ahmed Laatabi, Benoit Gaudou, Chihab Hanachi, Patricia Stolf, and Sébastien Truptil. Coupling agent-based simulation with optimization to enhance population sheltering. In *19th Information Systems for Crisis Response and Management Conference (ISCRAM 2022)*, pages à–paraître, 2022.

[10] Ngoc Luyen Le, Jinfeng Zhong, Elsa Negre, and Marie-Hélène Abel. Constraint-based recommender system for crisis management simulations. In *56th Hawaii International Conference on System Sciences, HICSS 2023*, 2023.

[11] Sonia Mehla and Sarika Jain. An ontology supported hybrid approach for recommendation in emergency situations. *Annals of Telecommunications*, 75(7) :421–435, 2020.

[12] Fushen Zhang, Shaobo Zhong, Simin Yao, Chaolin Wang, and Quanyi Huang. Ontology-based representation of meteorological disaster system and its application in emergency management : illustration with a simulation case study of comprehensive risk assessment. *Kybernetes*, 2016.