Actes des 36es journées francophones d'Ingénierie des Connaissances

Fleur Mougin

AfIA

Association française
pour l'Intelligence Artificielle

IC

36es journées francophones d'Ingénierie des Connaissances

PFIA 2025

Table des matières

Éditorial

36es 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 caractéristiques 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 systèmes "intelligents et explicables", capables d'aider l'humain dans ses activités et pour la prise 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, la conférence IC a reçu 25 soumissions d'articles. 17 articles ont été acceptés répartis dans les catégories suivantes : 8 articles longs, 6 articles courts et 3 articles déjà publiés dans une conférence internationale de renom. Un travail conséquent a été mené par les membres du comité de programme, chaque article a reçu entre 3 et 4 relectures comportant des critiques argumentées et constructives pour les auteurs.

Pour cette édition 2025, nous avons l'honneur et le plaisir d'accueillir Marieke van Erp, directrice du Digital Humanities Research Lab du KNAW Humanities Cluster aux Pays-Bas, dont la conférence invitée est intitulée "Layering Knowledge to Unpack the Layers of Meaning in Historical Texts".

Le programme de la conférence, réparti sur 2 jours et demi, suit un programme découpé en 6 sessions dont le contenu est détaillé dans ces actes. Les sessions portent sur des thèmes qui sont au cœur de l'ingénierie des connaissances : "Graphes de connaissances : conception et exploitation", "IA hybride et intégration neuro-symbolique", "Conception d'ontologies", "Modélisation et vérification formelle dans des contextes industriels", "Cadres ontologiques" et "Ingénierie des connaissances pour les humanités numériques", qui sera une session commune avec la journée Humanités numériques et IA.

Enfin, je voudrais remercier chaleureusement le comité de pilotage d'IC pour ses conseils avisés et les membres du comité de programme pour leur implication et leur contribution au succès de cette édition 2025 de la conférence IC. J'adresse également de vifs remerciements au comité d'organisation de la plateforme PFIA 2025 qui a été d'une aide précieuse.

Fleur Mougin

Comité de programme

Présidence

— Fleur Mougin, BPH - Inserm U1219, Université de Bordeaux.

Membres

— Xavier Aimé, Cogsonomy ;
— Nathalie Abadie, LaSTIG, IGN France ;
— Marie-Hélène Abel, Heudiasyc, Université de Technologie de Compiègne ;
— Yamine Ait Ameur, IRIT, INPT-ENSEEIHT ;
— Nathalie Aussenac-Gilles, IRIT ;
— Bruno Bachimont, Heudiasyc, University de Technologie de Compiègne ;
— Nacéra Bennacer-Seghouani, LRI, Centrale Supélec ;
— Nathalie Bricon-Souf, IRIT, Université Toulouse 3 Paul Sabatier ;
— Sandra Bringay, LIRMM, Université Paul-Valéry Montpellier ;
— Davide Buscaldi, LIPN, Université Sorbonne Paris Nord ;
— Sylvie Calabretto, LIRIS, INSA de Lyon ;
— Pierre-Antoine Champin, LIRIS, Université Claude Bernard Lyon 1 ;
— Jean Charlet, AP-HP & Inserm U1142 ;
— Victor Charpenay, LIMOS, MINES Saint-Etienne ;
— Adrien Coulet, HeKA - Inserm & Inria, Université Paris Cité ;
— Jérôme David, mOeX - LIG & Inria, Université Grenoble Alpes ;
— Sylvie Despres, LIMICS, Université Sorbonne Paris Nord ;
— Gayo Diallo, BPH - Inserm U1219, Université de Bordeaux ;
— Catherine Faron, I3S, Université Côte d'Azur ;
— Béatrice Fuchs, LIRIS, Université Jean Moulin Lyon III ;
— Frédéric Fürst, MIS, Université de Picardie Jules Verne ;
— Alban Gaignard, Institut du Thorax, Nantes Université ;
— Mounira Harzallah, LS2N, Nantes Université ;
— Nathalie Hernandez, IRIT, Université de Toulouse 2 Jean Jaurès ;
— Liliana Ibanescu, INRAE, AgroParisTech ;
— Sébastien Iksal, LIUM, Le Mans Université ;
— Antoine Isaac, Europeana & Vrije Universiteit Amsterdam ;
— Clément Jonquet, INRAE & LIRMM ;
— Mouna Kamel, IRIT, Université de Perpignan Via Domitia ;
— Gilles Kassel, Université de Picardie Jules Verne ;
— Maxime Lefrançois, LIMOS, MINES Saint-Etienne ;
— Dominique Lenne, Heudiasyc, Université de Technologie de Compiègne ;
— Pascal Molli, LS2N, Nantes Université ;
— Jérôme Nobécourt, LIMICS, Université Sorbonne Paris Nord ;
— Nathalie Pernelle, LIPN, Université Sorbonne Paris Nord ;
— Cédric Pruski, Luxembourg Institute of Science and Technology ;
— Joe Raad, LISN, Université Paris Saclay ;
— Sylvie Ranwez, EuroMov DHM, Ecole des Mines d'Alès ;
— Catherine Roussey, INRAE ;
— Fatiha Saïs, LISN, Université Paris Saclay ;
— Karim Sehaba, LIRIS, Université Lumière Lyon 2 ;
— Lina F. Soualmia, LITIS & LIMICS, Normandie Universités ;
— Konstantin Todorov, LIRMM, Université de Montpellier ;
— Cassia Trojahn, mOeX - LIG & Inria, Université Grenoble Alpes ;
— Raphaël Troncy, EURECOM ;
— Danai Symeonidou, INRAE ;
— Haïfa Zargayouna, LIPN, Université Sorbonne Paris Nord.

Session 1 : Graphes de connaissances - conception et exploitation

NutriKG – Un Graphe de Connaissances pour Modéliser les Préférences et les Besoins Nutritionnels

Alexandre Combeau[1,2], Fatiha Saïs [2], Nageeta Kumari[3], Stéphane Dervaux [1],
Cristina Manfredotti [1], Vincent Guigue [1], Paolo Viappiani [4]

[1] Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay, 91120, Palaiseau, France
[2] LISN, Université Paris-Saclay, CNRS, Gif-sur-Yvette, France
[3] Université Paris Saclay, Gif-sur-Yvette, France
[4] LAMSADE, CNRS and Université Paris-Dauphine, PSL, Paris, France

alexandre.combeau@universite-paris-saclay.fr

Résumé

Dans un contexte où la nutrition est essentielle à la prévention des maladies chroniques, il est crucial d'adapter les recommandations alimentaires aux besoins individuels. Pour répondre à cette complexité, nous avons développé NutriKG*, un graphe de connaissances intégrant les données des études INCA2 et INCA3 avec une ontologie détaillée. Cette approche permet d'inférer de nouvelles informations et de combler les lacunes des données existantes. L'intégration de règles SWRL et de schémas SHACL assure la cohérence des recommandations et leur explicabilité. Ainsi, l'utilisation de* NutriKG *permettrait de faciliter la génération de recommandations alimentaires précises, personnalisées et scientifiquement fondées.*

Mots-clés

Graphes de Connaissances, Ontologies, Recommandation Alimentaire, Personnalisation, Explicabilité.

Abstract

In a context where nutrition is essential for preventing chronic diseases, it is crucial to tailor dietary recommendations to individual needs. To address this complexity, we developed NutriKG*, a knowledge graph integrating data from the INCA2 and INCA3 studies with a detailed ontology. This hybrid approach enables the inference of new information and helps bridge gaps in existing datasets. The integration of SWRL rules and SHACL schemas ensures the consistency and explainability of recommendations. Thus,* NutriKG *facilitates the generation of precise, personalized, and scientifically grounded dietary recommendations.*

Keywords

Knowledge Graphs, Ontologies, Food Recommendation, Personnalization, Explainability.

1 Introduction

Dans un contexte de préoccupations croissantes en matière de santé publique, notamment l'obésité, le diabète et les maladies cardiovasculaires, l'importance d'une alimentation saine est largement reconnue comme un facteur clé pour prévenir et atténuer ces problèmes [11]. Toutefois, les habitudes alimentaires sont influencées par de nombreux facteurs, notamment les préférences individuelles, les contraintes professionnelles, le niveau d'activité physique et les contextes culturels, rendant difficile l'adoption d'un régime alimentaire universellement efficace [3].

Face à cette diversité, il devient impératif de proposer des approches personnalisées fondées sur une analyse approfondie des habitudes alimentaires et des besoins spécifiques de chaque individu [12]. L'automatisation de cette tâche en combinant des méthodes d'apprentissage automatique et des méthodes issues du Web sémantique avec les graphes de connaissances offrirait une solution efficace et explicable [9]. Un système de recommandation alimentaire, exploitant ces méthodes et technologies, pourrait générer des suggestions diététiques adaptées aux profils nutritionnels, aux antécédents médicaux et aux préférences alimentaires des utilisateurs [15].

L'intégration de données issues de multiples sources telles que les journaux alimentaires, les dossiers médicaux et les choix de mode de vie permettrait d'améliorer la précision et la pertinence des recommandations [16]. En fournissant des conseils diététiques scientifiquement fondés et personnalisés, un tel système pourrait favoriser des habitudes alimentaires plus saines et contribuer à la réduction de l'impact des maladies liées à la nutrition sur les systèmes de santé [2].

L'objectif d'une telle approche est d'accompagner les individus vers des choix alimentaires éclairés, favorisant un meilleur état de santé général et une augmentation de la qualité de vie. À plus grande échelle, ces recommandations personnalisées pourraient réduire la prévalence des maladies chroniques et améliorer la productivité globale de la société [10].

Pour développer un tel système, nous nous sommes appuyés sur les données de référence de l'étude INCA (*Étude Individuelle Nationale des Consommations Alimentaires*),

mises à disposition par l'ANSES (*Agence nationale de sécurité sanitaire de l'alimentation, de l'environnement et du travail*). Ce jeu de données constitue une ressource précieuse pour le développement d'outils tels que les systèmes de recommandation pour la nutrition. Bien que ces données soient bien structurées et de haute qualité, leur volume reste limité.

Dans [6], nous avons exploité les données INCA2 [1] afin d'entraîner un modèle capable de capturer les habitudes alimentaires des individus et de générer des recommandations de menus sous forme de séquences de plats. Dans un premier temps, nous avons présenté un modèle de recommandation inspiré d'une approche de type filtrage collaboratif. Cette première approche a montré ses atouts pour la recommandation d'un repas de type petit déjeuner, mais elle a révélé ses limites sur les repas plus complexes comme le déjeuner et le dîner. Nous avons présenté, ensuite, une seconde approche qui modélise le caractère séquentiel du déjeuner et du dîner et qui permet d'améliorer la performance de la recommandation sur ces repas. Cet approche repose sur une architecture de réseaux de neurones récurrents (*Recurrent Neural Networks*, RNN) pour l'apprentissage du modèle et la génération des séquences, en tenant compte de divers contextes (par exemple, les repas petit-déjeuner, déjeuner, dîner ou encore la tranche d'âge des individus). Ce travail a notamment démontré que l'intégration du contexte de consommation dans la génération d'une recommandation améliore considérablement sa pertinence.

Afin d'améliorer la précision du système de recommandation, de pallier la rareté des données et de garantir des recommandations explicables, nous avons développé *NutriKG*, un graphe de connaissances structuré en deux parties : (i) une composante conceptuelle, représentée par une ontologie, et (ii) une composante instancielle, construite à partir des données issues des deux études *INCA2* et *INCA3* [2] en 2006-2007 et en 2014-2015, respectivement.

Le développement du graphe de connaissances s'est déroulé en quatre étapes principales : (i) la création d'une première version de l'ontologie focalisée sur les consommations, leur structuration et leur composition ; (ii) l' extension de l'ontologie avec la modélisation des connaissances concernant les individus, leur contraintes et préférences alimentaires ; (iii) l'évolution de l'ontologie pour la prise en compte de l'organisation des données de la source INCA3 qui se différencient des données INCA2 par l'absence de séquentialité journalière des consommations et enfin (iv) la création des graphes de données conformes à l'ontologie et intégrant à la fois les données INCA2 et INCA3.

Ce graphe de connaissances est également enrichi par un ensemble de règles SWRL (Semantic Web Rule Language) [3] et de schéma de validation SHACL (Shapes Constraint Language) [4] pour compléter les informations manquantes (e.g., il manque l'information explicite sur le régime alimentaire pour 92% des individus dans INCA2), pour vérifier la conformité d'une recommandation par rapport aux contraintes de l'individu et enfin pour pouvoir fournir une explication intelligible pour les utilisateurs.

L'organisation de cet article est la suivante : la section 2 présente les travaux connexes, suivie par la section 3, qui introduit les préliminaires ainsi que les jeux de données issus de l'état de l'art. Les contributions principales sont détaillées dans la section 4. Ensuite, une première preuve de concept ainsi que les premiers résultats d'évaluation basés sur des questions de compétences sont présentés dans la section 5. Enfin, la section 6 conclut cet article et propose plusieurs perspectives pour de futurs travaux.

2 Travaux connexes

L'utilisation des ontologies dans le domaine alimentaire permet une modélisation formelle des connaissances relatives aux aliments, à leur composition, à leur transformation, ainsi qu'à leurs interactions avec la nutrition et la santé. Ces représentations sémantiques facilitent la structuration, le partage et l'interopérabilité des données. Plusieurs initiatives ont ainsi vu le jour pour formaliser et organiser ces connaissances sous la forme d'ontologies et de graphes de connaissances pour différents services et applications des domaines agroalimentaire et en santé.

Vocabulaires. Parmi ces initiatives, AGROVOC [13] constitue l'un des vocabulaires contrôlés les plus largement utilisés. Maintenu par la FAO (Organisation des Nations unies pour l'alimentation et l'agriculture), il couvre un large spectre de domaines, allant de l'alimentation et de la nutrition à l'agriculture, la pêche, la foresterie et l'environnement. Structuré sous forme de thésaurus hiérarchique, AGROVOC permet une organisation fine des concepts et favorise l'interopérabilité avec d'autres systèmes de gestion de l'information. Il est notamment utilisé pour l'indexation et la recherche d'informations dans le domaine agroalimentaire.

Ontologies des aliments. D'autres ontologies, plus spécifiques, se concentrent sur la représentation des aliments eux-mêmes et de leurs propriétés. FoodOn [4], intégrée à l'OBO Foundry [5], vise à décrire les entités alimentaires sous différents aspects : classification des aliments, composition nutritionnelle, propriétés physiques et rôle dans la chaîne alimentaire. Elle joue un rôle clé dans la structuration des données liées à la sécurité alimentaire, à la nutrition et à l'agriculture. FoodOn inclut également des catégories permettant de représenter les ingrédients, les produits transformés, ainsi que leurs relations avec les pratiques agricoles et les réglementations alimentaires. Cette approche facilite l'analyse des données et leur intégration dans des systèmes d'aide à la décision pour la gestion de la qualité et de la traçabilité des aliments.

Qualité et traçabilité. Outre la modélisation des aliments eux-mêmes, certaines ontologies ont été spécifiquement conçues pour l'évaluation des risques alimentaires. C'est notamment le cas de l'ontologie développée par l'ANSES

1. INCA2 : https://tinyurl.com/3zxjm6ca
2. INCA3 : https://tinyurl.com/487pj39d
3. SWRL : https://www.w3.org/submissions/SWRL/
4. SHACL : https://www.w3.org/TR/shacl/
5. https://obofoundry.org/

[8], qui structure les informations relatives aux expositions et aux dangers alimentaires. En s'appuyant sur l'alignement d'ontologies existantes, elle modélise des concepts tels que les contaminants alimentaires, les seuils de toxicité, les groupes à risque et les effets sanitaires potentiels. L'objectif est d'améliorer la précision des évaluations de risque et de permettre une automatisation partielle des analyses grâce aux raisonnements ontologiques.

Un autre axe de recherche concerne les ontologies dédiées aux processus de transformation des aliments. Certaines, comme celle développée par [7], modélisent les différentes étapes de préparation, cuisson et conservation des aliments. Ces ontologies intègrent des connaissances sur l'impact des procédés de transformation (e.g., fermentation, pasteurisation, surgélation) sur les qualités nutritionnelles et sanitaires des aliments. Elles décrivent également l'évolution des propriétés organoleptiques des aliments en fonction des traitements subis. Ces représentations sont particulièrement utiles pour les systèmes de traçabilité alimentaire et l'optimisation des procédés industriels, contribuant ainsi à une meilleure gestion de la qualité des produits transformés.

Dans le secteur de la production animale, l'ontologie développée dans le cadre du projet européen INTAQT [6] se focalise sur la viande de poulet, de boeuf et de produits laitiers. Elle définit et structure les concepts liés à la qualité, à la traçabilité et à la production de la viande. Elle intègre des données sur les races bovines, les méthodes d'élevage, les standards de qualité ainsi que les critères sensoriels tels que le goût, la texture et la couleur. Cette ontologie vise à améliorer l'interopérabilité des données entre les différents acteurs de la chaîne agroalimentaire et à renforcer la traçabilité des produits carnés et laitiers.

Ontologies et systèmes de recommandation en nutrition. Enfin, l'intégration des données alimentaires dans des systèmes de recommandation personnalisée constitue un défi majeur. FoodKG [5] est un graphe de connaissances conçu pour répondre à cet enjeu en combinant plusieurs sources de données alimentaires (i.e., ontologies, bases de données de recettes, publications scientifiques). Il permet d'exploiter des connaissances sémantiques pour affiner les recommandations alimentaires en fonction des préférences des utilisateurs, de la disponibilité des ingrédients et des contraintes nutritionnelles. Grâce à l'utilisation de SPARQL, FoodKG facilite l'interrogation et l'extraction d'informations pour générer des suggestions alimentaires plus pertinentes et adaptées aux consommateurs.

Ces différentes ontologies illustrent l'étendue des travaux menés pour structurer et modéliser les connaissances alimentaires. Elles constituent une base essentielle pour l'intégration et l'exploitation des données dans des systèmes d'information, des applications en nutrition et des services de recommandation alimentaire. Notre travail sur NutriKG est une extension de certaines de ces ressources existantes tel que FoodKG en fournissant un graphe de connaissance capturant davantage les comportements et les habitudes alimentaires des individus recueillis grâce aux initiatives INCA2 et INCA3.

6. https://www.sysaaf.fr/les-programmes-de-r-d/programmes-r-d-avi-en-cours/intaqt

3 Préliminaires

3.1 Ontologies et graphes de connaissances

Pour représenter sémantiquement les connaissances liées aux consommations nutritionnelles nous avons eu recours aux *ontologies* et aux *graphes de connaissances*. Une ontologie peut être définie comme une représentation formelle et structurée des connaissances d'un domaine, définissant des concepts, leurs relations et leurs propriétés, afin de permettre une compréhension partagée et une exploitation par des machines. On considère un graphe de connaissances comme une structure de données qui organise l'information sous forme de nœuds (entités) et d'arêtes (relations), facilitant l'intégration, le raisonnement et l'extraction de connaissances à partir de sources hétérogènes.

Dans la définition 3.1, nous donnons une définition formelle d'un graphe de connaissances et des éléments de l'ontologie qui permettent de le structurer.

Définition 3.1. (Graphe de connaissances RDF). Nous considérons un graphe de connaissances RDF défini par un couple $(\mathcal{O}, \mathcal{G})$, où :

– $\mathcal{O} = (\mathcal{C}, \mathcal{P})$ est une ontologie représentée en OWL2 [7] 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 faits représenté par des triplets de la forme $\{(\text{sujet, propriété, objet}) \mid \text{sujet} \in \mathcal{I}, \text{propriété} \in \mathcal{P}, \text{objet} \in \mathcal{I} \cup \mathcal{L}\}$, où $\mathcal{I}$ est l'ensemble d'instances de classes $c \in \mathcal{C}$ désignées par des IRI (Internationalized Resource Identifier), $\mathcal{P}$ est l'ensemble des propriétés, et $\mathcal{L}$ est l'ensemble des littéraux (tels que les nombres et les chaînes de caractères). On notera $I_C \subseteq \mathcal{I}$ l'ensemble d'instances de la classe $C \in \mathcal{C}$.

Les ontologies peuvent être enrichies par des règles, comme celles exprimées en SWRL, afin d'inférer de nouvelles connaissances à partir des faits existants. SWRL permet de définir des règles en logique du premier ordre pouvant exprimer le fait d'identifier qu'un aliment contenant un ingrédient allergène doit être évité par un individu. L'intégration de ces règles permet d'inférer de nouvelles connaissances, d'affiner les recommandations ou encore de détecter des incohérences.

3.2 Les jeux de données et référentiels

La construction du graphe de connaissances nutritionnelles repose sur l'intégration de plusieurs ensembles de données. Parmi ceux-ci, les principales sources actuellement utilisées incluent les sources de données INCA2 et INCA3,

7. OWL : https://www.w3.org/TR/owl2-overview/

alignées respectivement aux référentiels CIQUAL et FoodEx2 [8]. Ces sources de données fournissent des informations essentielles sur les habitudes alimentaires, la composition nutritionnelle des aliments et leur classification.
En complément, l'expertise humaine est mobilisée pour enrichir le graphe par des classes et des relations du domaine nécessitant une validation ou une interprétation spécifique. Dans la suite nous présentons ces différentes sources de données et référentiels ainsi que leur exploitation pour la construction et l'évolution du graphe de connaissances.

3.2.1 Le jeu de données INCA2

Le jeu de données INCA2 est issu d'une enquête nationale visant à analyser les comportements alimentaires des individus âgés de 3 à 79 ans vivant en France métropolitaine. Il repose sur un échantillon de 4 079 participants, répartis en deux groupes : les enfants (3 à 17 ans) et les adultes (18 à 79 ans). L'étude prend en compte divers facteurs démographiques et socio-économiques, tels que la région, le sexe, la taille du ménage et l'âge des individus.
Pour les enfants, l'enquête recueille des informations spécifiques sur les habitudes alimentaires en dehors du domicile ainsi que sur les préférences alimentaires (p. ex. : consommation de lait, de fruits). Chez les adultes, des données complémentaires sont intégrées, notamment sur l'activité physique et la consommation de certains produits comme la cigarette.
Concernant la consommation alimentaire, INCA2 s'appuie sur un journal alimentaire couvrant sept jours consécutifs, où chaque individu renseigne l'ensemble de ses repas : petit-déjeuner, déjeuner, dîner, ainsi que les collations intermédiaires. En complément, un questionnaire permet de collecter des informations détaillées sur les facteurs socio-économiques et les habitudes de vie.
Ce jeu de données comprend 1 280 références alimentaires et boissons, chaque élément étant associé à un vecteur nutritionnel extrait de la base CIQUAL 2008, qui fournit des données précises sur la composition nutritionnelle des aliments.

3.2.2 Le jeu de données INCA3

L'enquête INCA3, menée entre 2014 et 2015, repose sur un échantillon plus large de 5 855 participants, comprenant 2 698 enfants (0 à 17 ans) et 3 157 adultes (18 à 79 ans). Contrairement à INCA2, qui s'appuyait sur un suivi alimentaire continu sur sept jours, INCA3 adopte une méthodologie différente en collectant les informations sur deux à trois journées de 24 heures non consécutives. Cette approche, bien que plus souple, rend difficile la comparaison directe des résultats entre les deux enquêtes, comme le souligne la documentation officielle d'INCA3.
INCA3 apporte également une amélioration notable en intégrant des données plus détaillées sur divers aspects : facteurs socio-économiques, habitudes alimentaires, activité physique, état de santé et préférences alimentaires des enfants. L'enquête permet ainsi une analyse plus fine et approfondie des comportements alimentaires.
Grâce à sa richesse et à son niveau de détail, INCA3 a été utilisé dans un second temps pour enrichir la partie de notre ontologie dédiée aux consommateurs. Son exploitation permet d'améliorer la personnalisation des recommandations alimentaires en prenant en compte les besoins nutritionnels individuels ainsi que les facteurs liés au mode de vie.

3.2.3 Le jeu de données FoodEx2

Le jeu de données FoodEx2 est un système de classification des aliments développé par l'Autorité européenne de sécurité des aliments (EFSA). Conçu pour fournir un cadre standardisé de catégorisation et de codage des produits alimentaires, il facilite la collecte, l'analyse et l'interopérabilité des données dans les domaines de la sécurité alimentaire et de la nutrition.
Ce système repose sur une structure hiérarchique détaillée, attribuant à chaque produit alimentaire une description précise et un code spécifique. Cette organisation garantit une identification rigoureuse des aliments et assure la comparabilité des données entre différentes études et bases de données. Grâce à sa précision et à sa flexibilité, FoodEx2 est largement utilisé dans la recherche scientifique, la réglementation et la surveillance de la santé publique, contribuant ainsi à améliorer la fiabilité des informations nutritionnelles et alimentaires.
Dans le cadre de l'enquête INCA3, les aliments recensés sont déjà associés aux codes FoodEx2, permettant une classification détaillée. Les données incluent des informations sur le groupe, le sous-groupe, le sous-sous-groupe ainsi que le code spécifique FoodEx2, enrichi de facettes permettant de préciser davantage les caractéristiques des produits alimentaires.

3.2.4 Le référentiel CIQUAL

Le référentiel CIQUAL [9] est une base de données développée par l'ANSES, fournissant des informations sur la composition nutritionnelle des aliments consommés en France. Il répertorie plusieurs centaines d'aliments avec des données sur les macronutriments, minéraux, vitamines et autres composés. CIQUAL fournit des données sur la composition de plusieurs centaines d'aliments, couvrant un large éventail de catégories alimentaires (produits bruts, transformés, plats préparés, etc.). Chaque aliment est décrit par une série de paramètres nutritionnels, incluant :

- Macronutriments : Protéines, lipides, glucides, fibres alimentaires
- Éléments minéraux : Calcium, fer, magnésium, sodium, etc.
- Vitamines : Vitamine C, vitamines du groupe B, vitamine A, etc.
- Autres composés : Acides gras, sucres, additifs alimentaires

Les données proviennent d'analyses en laboratoire, de l'industrie agroalimentaire et de sources scientifiques. CIQUAL est mis à jour régulièrement et est utilisé en épidémiologie nutritionnelle, pour l'évaluation des apports alimentaires, le développement de produits et la modélisation

8. FoodEx2 :https://agroportal.lirmm.fr/ontologies/FOODEX2

9. https://ciqual.anses.fr/

des risques.

4 Le graphe de connaissances NutriKG

Dans cette section nous décrivons les éléments principaux qui composent le graphe de connaissances NutriKG ainsi que la méthodologie de sa construction. Nous présentons tout d'abord l'ontologie puis la méthodologie de construction de NutriKG.

4.1 L'ontologie NutriKG

Comme le montre la Figure 1, l'ontologie modélise deux aspects majeurs de la consommation alimentaire. La partie supérieure représente la composition des consommations, en mettant l'accent sur les aliments et les nutriments, tandis que la partie inférieure est dédiée à la modélisation des individus, intégrant leurs préférences, ainsi que leurs contraintes sanitaires et personnelles.

A) Consommations – classes et relations. Une consommation est modélisée à travers plusieurs classes interconnectées. La classe centrale, `FullDayConsumption`, représente l'ensemble des repas consommés par un individu au cours d'une journée (i.e., petit-déjeuner, déjeuner, collation et dîner). Cette classe est liée à `FoodComposition`, qui décrit pour chaque repas l'ensemble des plats consommés.

Les aliments consommés peuvent être simples, comme *eau* ou *banane*, ou composés, comme *tarte au citron*. Chaque aliment est relié à sa classification `foodex2`, permettant d'accéder à l'ensemble des informations fournies par FoodEx2, notamment la composition nutritionnelle et les différentes facettes associées à un plat (e.g., *mode de préparation, origine géographique*).

À chaque aliment est également associée sa composition nutritionnelle, obtenue par l'alignement des données INCA2 à CIQUAL. Les quantités des nutriments sont représentées sous forme de valeurs numériques accompagnées de leurs unités de mesure.

Un élément clé de la modélisation concerne la séquentialité des aliments consommés au sein d'un repas et des repas successifs au fil des jours. Cette dimension, issue des données INCA2 et INCA3, est intégrée dans l'ontologie à travers plusieurs propriétés temporelles et relationnelles. Celles-ci permettent de représenter à la fois l'ordre de consommation des aliments au sein d'un même repas et la chronologie des consommations journalières.

Parmi ces propriétés, la classe `FullDayConsumption` possède la propriété `before`, une propriété de type objet qui relie une instance de `FullDayConsumption` à une autre, correspondant à la consommation du jour précédent (qui ne correspond pas nécessairement au jour immédiatement antérieur).

En complément, les propriétés `hasBeginning` et `hasDuration` permettent de modéliser la séquence des aliments au sein d'un même repas ainsi que la durée associée à la consommation de chaque partie du repas (e.g., entrée, plat principal, boisson, dessert).

B) Individus – classes et relations. Dans cette seconde partie de l'ontologie, nous modélisons les caractéristiques des individus pouvant influencer la recommandation alimentaire.

Tout d'abord, la classe `Individu` regroupe des informations générales telles que la tranche d'âge, l'indice de masse corporelle (BMI), le genre ainsi que son profil alimentaire où l'on peut retrouver des informations sur son régime alimentaire (e.g., `VeganDiet`, `KetoDiet`, `LooseWeightDiet`)

Ensuite, deux classes permettent de représenter les préférences alimentaires de l'individu. La classe `FoodPreferences` modélise l'attrait d'une personne pour un aliment avec un booléen. En complément, la classe `FoodInterests` capture les préférences liées aux modes de préparation des repas (e.g., faits maison) ainsi que la propension de l'individu à découvrir de nouveaux aliments.

Les aspects médicaux et les contraintes de santé sont également pris en compte. La classe `MedicalInformation` regroupe des informations telles que le poids, la taille, la consommation de tabac et la volonté de perdre du poids. Quant à la classe `Restrictions`, elle modélise les éventuelles restrictions médicales ou allergies alimentaires de l'individu (e.g., gluten, fruits de mer).

Enfin, les habitudes et contraintes personnelles sont représentées par deux classes supplémentaires. La classe `PhysicalActivity` décrit les niveaux d'activité physique de l'individu, tandis que la classe `FoodProfile` formalise ses comportements alimentaires ainsi que ses objectifs liés à la gestion du poids.

4.2 NutriKG : méthodologie de construction

Nous avons construit l'ontologie NutriKG en suivant le 2ème scénario de la méthodologie de construction d'ontologie NeOn [14], c'est-à-dire celui qui consiste en la réutilisation et la réingénierie de ressources non ontologiques (NOR). La construction de l'ontologie en Turtle (TTL) a été réalisée en utilisant Chowlk [1], un outil permettant de convertir une modélisation UML en RDF. Cette approche assure une traduction fidèle du modèle conceptuel en un format exploitable pour le web sémantique.

Une première version de l'ontologie et du graphe de connaissances a été conçue à partir des descriptions de consommations issues du jeu de données INCA2. Par la suite, en s'appuyant sur les informations détaillées dans INCA3, tant sur les habitudes de consommation que sur les caractéristiques des individus, cette ontologie initialement centrée sur la consommation a été enrichie. De nouveaux concepts et relations ont été intégrés afin d'inclure une représentation plus complète des individus et de leurs profils associés, renforçant ainsi la capacité du graphe à modéliser les interactions entre les consommateurs et leurs choix alimentaires.

Afin de maximiser la quantité de données disponibles pour l'entrainement du modèle du système de recommandation, nous avons fixé comme objectif la conception d'une ontologie unificatrice capable d'intégrer de manière homogène

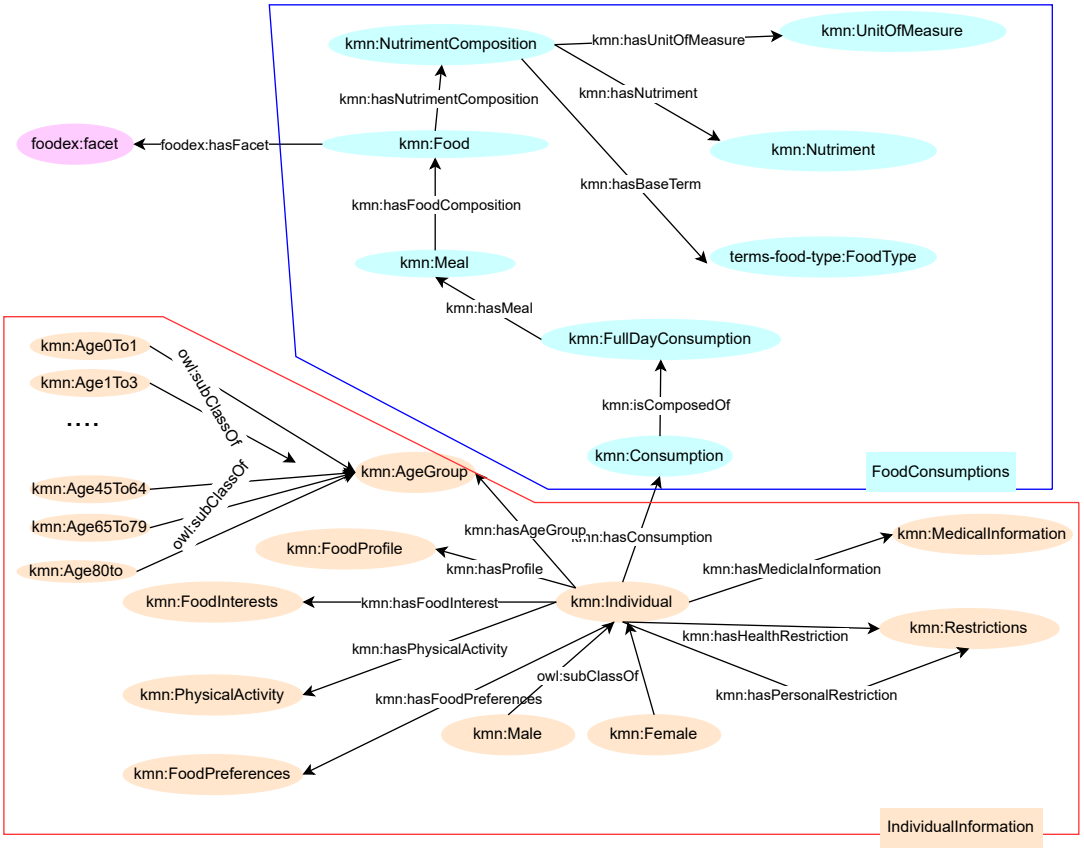

FIGURE 1 – L'ontologie du graphe de connaissances NutriKG issu des études INCA2 et de INCA3.

(sans perte d'informations) les jeux de données INCA2 et INCA3. Pour y parvenir, nous avons dû opter pour des choix de modélisation permettant de gérer l'hétérogénéité inhérente à ces sources de données. Cette hétérogénéité repose principalement sur deux aspects clés :

Période de collecte des données. Dans INCA2, la consommation alimentaire est enregistrée sur 7 jours consécutifs pour chaque individu, tandis que dans INCA3, seuls 3 jours non consécutifs sont pris en compte. Cette différence méthodologique a constitué un défi majeur dans la conception d'une ontologie capable de concilier ces deux approches et d'intégrer efficacement ces données.

Nomenclature et cartographie. Les bases INCA2 et INCA3 reposent sur des nomenclatures distinctes. Dans INCA3, la nomenclature a été associée à la classification FoodEx2. Ce dernier introduit des facettes, permettant d'affiner la description des aliments. Par conséquent, un même aliment dans INCA3 peut apparaître sous plusieurs déclinaisons, compliquant l'alignement avec INCA2. Le nombre exact d'instances présentes dans ces deux jeux de données est détaillé ci-dessous.

Dans le tableau 1 nous présentons quelques statistiques descriptives des deux jeux de données INCA2 et INCA3, i.e., nombre d'individus, nombre de jours, nombre de consommation et la taille d'une consommation.

Dataset	#Individus	#Jours	#Conso	Taille conso
INCA2	4079	7	80052	5.8
INCA3	3900	3	34964	5.7

TABLE 1 – Descriptions des données INCA

4.2.1 Homogénéisation des périodes de collecte des données

En raison des différences dans les méthodologies de collecte des données alimentaires entre INCA2 et INCA3, nous avons rencontré une incohérence structurelle dans la représentation des consommations. Pour y remédier, plusieurs ajustements ont été apportés à l'ontologie afin d'assurer une intégration plus cohérente et une meilleure exploitation des données.

La Figure 1 illustre la dernière version de l'ontologie mise à jour après résolution des incohérences. Dans la première version, un consommateur était associé à une consommation hebdomadaire, laquelle englobait des consommations journalières. Cette structuration a été revue afin de mieux correspondre à la nature des données collectées, notamment dans INCA3 où les consommations ne sont pas enregis-

trées sur une semaine complète. Ainsi, comme le montre la Figure 1, un consommateur est désormais associé à des consommations journalières indépendantes, sans contrainte temporelle spécifique à une semaine donnée.
Par ailleurs, pour mieux gérer l'historique des consommations, nous avons intégré l'ontologie Time (voir Figure 1). Cette amélioration est particulièrement pertinente pour INCA3, où les jours de consommation sont non consécutifs. Grâce à ce mécanisme, le système peut suivre les consommations passées et exploiter ces données pour formuler des recommandations alignées sur les habitudes alimentaires antérieures des utilisateurs, favorisant ainsi une alimentation plus équilibrée et personnalisée.
Enfin, nous avons restructuré la classification des aliments et des denrées afin d'adopter une hiérarchie conforme aux codes FoodEx2. Cette nouvelle organisation garantit une meilleure correspondance entre les différentes sources de données et facilite l'intégration des informations nutritionnelles dans le graphe de connaissances.

4.2.2 Homogénéisation des nomenclatures

Étant donné qu'INCA3 est déjà aligné sur FoodEx2, un standard largement adopté dans le domaine, nous avons choisi d'établir également une correspondance entre INCA2 et FoodEx2. Cependant, cette tâche a soulevé une difficulté majeure : les libellés des aliments dans INCA2 sont en français, tandis que ceux de FoodEx2 sont en anglais. Pour surmonter cet obstacle, nous avons exploré plusieurs options de traduction et constaté que l'API DeepL Translator [10] offrait les meilleurs résultats. Nous avons donc utilisé cette solution pour traduire les noms des aliments, des groupes alimentaires et des sous-groupes alimentaires d'INCA2 en anglais.
Afin d'automatiser la mise en correspondance entre INCA2 et FoodEx2, nous avons exploité l'application de codage intelligent FoodEx2. Cet outil, basé sur des réseaux neuronaux et des modèles Scapy [11], permet d'associer n'importe quel aliment à un code et à des facettes FoodEx2. Son code étant librement accessible sur GitHub, il constituait une solution adaptée à nos besoins. Cette approche a déjà été validée dans une étude précédente visant à établir une correspondance entre les produits alimentaires suédois et les codes FoodEx2.
Nous avons utilisé le modèle BaseTerm (BT) avec un seuil de similarité pour associer les aliments d'INCA2 aux codes FoodEx2. Les paramètres et les résultats détaillés de cette mise en correspondance sont présentés dans le tableau ci-dessous.
Dans le tableau 2 nous montrons les résultats des deux expériences menées. Dans la première, nous avons appliqué le modèle BT avec un seuil de 40 % (après plusieurs tests) en utilisant les libellés traduits des aliments (Libal), ce qui a permis de faire correspondre 1 196 instances de la nomenclature INCA2 sur un total de 1 343, laissant 147 instances non appariées. Afin d'améliorer ce résultat, nous avons exploité les informations sur les groupes et sous-groupes alimentaires dans une seconde expérience. Cette fois, nous avons utilisé le libellé du groupe alimentaire INCA2 (Libal_gr) avec le même modèle BT et le même seuil, ce qui nous a permis d'apparier 113 des 147 instances restantes.
L'objectif étant d'aligner les termes de base avec FoodEx2, cette approche s'est révélée efficace, ne laissant que 34 aliments non cartographiés, qui ont été traités manuellement à l'aide de connaissances expertes.

Carac. INCA2	Ins. mappées	Ins. Restantes
Libal	1196	147
Libal_gr	113	34

TABLE 2 – Alignement de INCA2 et Foodex2

5 Preuve de concept et évaluation

Dans cette section nous présentons nos résultats préliminaires de l'évaluation du graphe de connaissances NutriKG en nous appuyant sur des questions de compétences et sur trois applications possibles : (i) utilisation de schémas SHACL pour la vérification de la conformité des recommandations produites, (ii) l'utilisation du graphe et des règles SWRL pour inférer des informations manquantes, et (iii) production d'explications pour une recommandation de donnée.
L'ontologie de *NutriKG* a été mise à disposition de la communauté via *recherche.data.gouv.fr* et accessible rendu accessible avec le DOI https://doi.org/10.57745/037D7N. Les données du graphe de connaissances sont mises à disposition via un SPARQL endpoint généré par le triple store GraphDB [12].

5.1 Evaluation de l'ontologie et du graphe de connaissances

L'évaluation de l'ontologie est réalisée grâce à une série de questions de compétences issues des sites officiels de INCA 2 et de INCA 3 et un autre ensemble de questions fournies par les experts du domaine.

Q1	*Quelle quantité de nourriture mangent chaque jour les Français ?*
Q2	*Quel est le pourcentage d'obésité dans la population adulte ?*
Q3	*Quel est nombre d'individus déclarés végétariens ?*
Q4	*Quel est nombre d'individus allergiques aux oeufs ?*
Q5	*Quel est nombre d'individus n'ayant pas déclarés de régime alimentaire ?*

Pour la question Q2, nos résultats sont similaires aux observations de l'ANSES pour INCA2. Cependant, le prétraitement des données a eu un impact plus marqué sur INCA3, en particulier en raison du choix de ne conserver que les

10. https://www.deepl.com/en/pro-api
11. https://scapy.net/
12. https://graphdb.ontotext.com/

Question	Res. INCA2	Res. INCA3	évolution
Q1	2939g	2122g	817g
Q2	10.5%	16.7%	+6.2%
Q3	23	0	- 23
Q4	0	6	+ 6
Q5	3788	3435	- 353

TABLE 3 – Résultats des questions de compétences et expertes sur INCA2 et INCA3

Dataset	$\geq$ 18ans	15 - 17ans	11 - 14ans	3 - 10ans
INCA2	2939	1955	1898	1766
INCA3	2122	1775	1703	1415

TABLE 4 – Quantité de nourriture consommé (aliments et boissons) par jour en grammes selon l'âge sur les données INCA

repas du petit-déjeuner, du déjeuner et du dîner. Cette différence pourrait également refléter une évolution des habitudes de consommation.

5.2 Restrictions et profils

Cette ontologie peut également être intégrée à un système de recommandation, permettant ainsi d'évaluer la qualité d'une recommandation de repas pour un utilisateur en vérifiant le respect des contraintes qui lui sont associées.

Afin de modéliser les différentes contraintes alimentaires liées à la santé ou aux préférences personnelles des individus présentes dans les données INCA, nous les avons représentée par les classes `kmn:FoodProfile` et `kmn:Restrictions` (voir Figure 2).

Les restrictions définissent des contraintes négatives sur les aliments, influençant défavorablement leur recommandation à un utilisateur. Deux types de liens sont distingués : `kmn:hasHealthRestriction` et `kmn:hasPersonalRestrictions`, qui déterminent la sévérité de la restriction.

Une préférence personnelle peut, dans de rares cas, être prise en compte dans une recommandation, tandis qu'une contrainte de santé est strictement interdite. Par exemple, un plat contenant des œufs peut être recommandé à une personne ayant une restriction personnelle sur cet aliment, mais jamais si cette personne y est allergique.

La modélisation des habitudes et préférences de consommation s'appuie sur les profils `kmn:FoodProfiles`, qui traduisent une appétence particulière pour certains groupes d'aliments. Bien que ces profils soient déjà présents dans les données INCA3, aucune description ne précise les biais qu'ils devraient induire. Ainsi, notre objectif est de formuler un ensemble de contraintes afin de représenter au mieux les données réelles et l'état des connaissances expertes sur ces habitudes alimentaires.

Ces contraintes peuvent être utilisées pour contraindre la recommandation d'un individu. Les restrictions alimentaires sont exprimées à l'aide de schémas SHACL. Ci-dessous, nous présentons un exemple simplifié d'une restriction concernant les œufs. L'idée est de récupérer les consommations d'un utilisateur et de vérifier qu'aucune d'elles ne contient d'œufs. Pour distinguer une contrainte de santé d'une préférence personnelle, nous utilisons `sh:severity`, qui déclenche une violation de contrainte lorsqu'une contrainte de santé est enfreinte, et uniquement un avertissement dans le cas d'une préférence.

```
@prefix kmn: <http://example.org/kmn#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .

kmn:NoEggRestrictionShape a sh:NodeShape ;
  sh:targetClass kmn:Individual ;
  sh:property [
    sh:path kmn:hasHealthRestriction ;
    sh:node kmn:NoEgg;] .
kmn:NoEggRestriction a sh:NodeShape ;
  sh:property [
    sh:path kmn:hasConsumption ;
    sh:node [
       sh:path kmn:hasFood ;
       sh:node [
         sh:path kmn:foodGroup ;
        sh:not [ sh:hasValue kmn:Egg ];] ;] ;
    sh:severity sh:Violation ;] .
```

Nous avons également enrichi le graphe de connaissances *NutriKG* avec un ensemble de règles en logique du premier ordre afin d'inférer et d'expliciter certaines connaissances sur les individus, auparavant implicites dans le graphe, telles que leur régime alimentaire. Ces règles, exprimées en SWRL, peuvent être exploitées par un raisonneur pour déduire de nouvelles connaissances.

Par exemple, un individu peut être défini comme végétarien s'il consomme une proportion de légumes supérieure à un seuil prédéterminé, ou à l'inverse, s'il consomme une quantité de produits d'origine animale inférieure à un seuil donné.

À titre d'exemple, et pour simplifier son expression, on peut formuler une règle en logique du premier ordre définissant les conditions à satisfaire pour qu'un individu soit considéré comme végétarien. Ainsi, pour un individu `ind` et ses consommations `foodC`, on définit une règle utilisant une fonction qui calcule la proportion de produits d'origine animale par rapport au nombre total de consommations, notée `countAnimal`. Si cette proportion dépasse un seuil `S`, on peut alors inférer le `FoodProfile` de type `VegetarianDiet`, que l'on associe à l'individu.

```
kmn:Individual(?ind) ^
kmn:hasProfile(?ind, ?profile)  ^
kmn:hasConsumption(?ind, ?foodC) ^
kmn:hasFood(?food, ?foodComp)^
kmn:foodGroup(?food, ?group)^
sqwrl:count(?totalCount, ?foodC)^
sqwrl:countDistinct(?aniCount,?foodC,
                ?group,?kmn:AnimalDerivedFood)^
swrlb:divide(?aniRatio,?aniCount,?totalCount)^
swrlb:greaterThanOrEqual(?aniRatio, S) =>
kmn:VegetarianDiet(?profile)
```

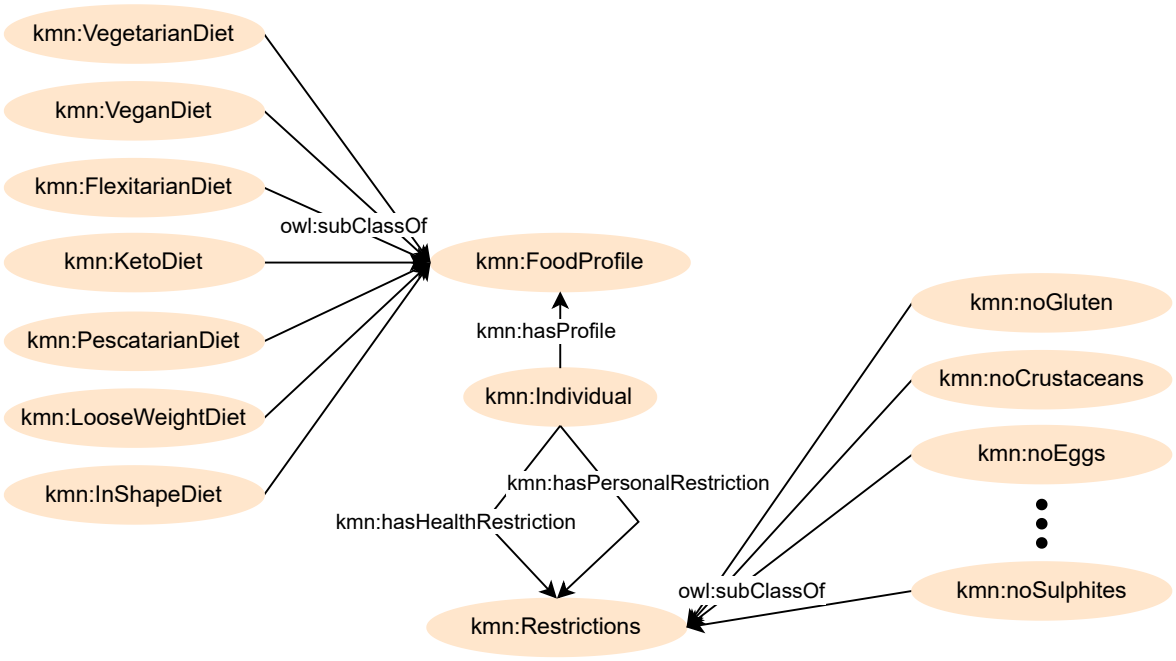

FIGURE 2 – NutriKg - Représentation des restrictions et profils alimentaires

À l'aide de ces profils, nous regroupons ensuite les aliments en fonction des profils des individus qui les consomment. Pour chaque aliment, nous calculons la proportion de profils qui l'incluent dans leur consommation. Cette analyse peut également être étendue à des classes et groupes d'aliments afin de faciliter la généralisation. Ce biais peut ensuite être exploité lors des recommandations, permettant ainsi au graphe d'enrichir les données préexistantes. Enfin, ces règles peuvent également servir de support pour la production d'explications pour les recommandations faites aux utilisateurs.

Au travers de cette première preuve de concept nous avons voulu montrer le potentiel de l'utilisation de la richesse du contenu de NutriKG dans le cadre de la recommandation alimentaire.

6 Conclusion et Travaux futurs

Dans un contexte où la nutrition est essentielle pour la prévention des maladies chroniques, la personnalisation des recommandations alimentaires est devenue un enjeu majeur. La diversité des facteurs influençant l'alimentation, tels que les préférences, les contraintes médicales et les habitudes culturelles, complexifie cette tâche. Dans cet article nous avons présenté NutriKG, un graphe de connaissances qui intègre des données de consommation issues des études INCA2 et INCA3, en les couplant à une ontologie, des règles SWRL et des schémas SHACL. Cette approche hybride associe l'organisation formelle des connaissances à la flexibilité des données réelles, ce qui permet non seulement d'inférer des informations manquantes, mais aussi de pallier les lacunes des jeux de données existants.

L'évaluation préliminaire de NutriKG montre la pertinence de la formalisation des connaissances et la structuration des données au travers plusieurs tâches : (i) réponses aux questions et en particulier à certaines questions de compétence de INCA2 et de INCA3 ; (ii) la capacité à inférer de nouvelles connaissances tels que les régimes alimentaires des individus, (iii) la vérification de conformité des recommandations vis-à-vis des restriction personnelles et médicales des individus grâce aux schémas SHACL et enfin (iv) à fournir des explications intelligibles pour des recommandations alimentaires.

Nous envisageons dans les travaux futurs plusieurs extensions de ce travail. Tout d'abord l'extension de l'alignement de l'ontologie NutriKG avec CIQUAL et d'autres ontologies existantes telles que FoodOn [4] ou encore celle de FoodKG[5]. L'intégration de NutriKG dans le système de recommandation tel que celui développé dans [6]. Cela peut être réalisé en post-traitement, d'abord, pour filtrer les recommandations incompatibles avec le profil/préférences de l'utilisateur, ensuite dans la phase d'apprentissage pour produire des recommandations plus pertinentes. Il serait également intéressant d'explorer la piste de l'utilisation du graphe de connaissances pour aider à l'augmentation de données. Enfin, nous souhaiterions proposer différentes définitions d'explications (e.g., contrastives, contrefactuelles) et des schémas de génération adaptés.

Remerciements

Cette recherche a été soutenue par l'Institut DATAIA Convergence dans le cadre du Programme d'Investissement d'Avenir (ANR-17-CONV-0003) opéré par l'Université Paris-Saclay et par la graduate school ISN (Informatique et Science du Numérique) de l'université Paris Saclay.

Références

[1] Serge Chávez-Feria, Raúl García-Castro, and María Poveda-Villalón. Chowlk : from uml-based ontology conceptualizations to OWL. In Paul Groth, Maria-Esther Vidal, Fabian M. Suchanek, Pedro A. Szekely, Pavan Kapanipathi, Catia Pesquita, Hala Skaf-Molli, and Minna Tamper, editors, *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*, volume 13261 of *Lecture Notes in Computer Science*, pages 338–352. Springer, 2022.

[2] GBD 2017 Diet Collaborators. Health effects of dietary risks in 195 countries, 1990–2017 : a systematic analysis for the global burden of disease study 2017. *The Lancet*, 393(10184) :1958–1972, 2019.

[3] Tom Deliens, Peter Clarys, Ilse De Bourdeaudhuij, and Benedicte Deforche. Determinants of eating behaviour in university students : a qualitative study using focus group discussions. *BMC Public Health*, 14(1) :53, 2014.

[4] H. 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, 2018.

[5] Steven Haussmann, Oshani Seneviratne, Yu Chen, Yarden Ne'eman, James Codella, Ching-Hua Chen, Deborah L. McGuinness, and Mohammed J. Zaki. Foodkg : A semantics-driven knowledge graph for food recommendation. In *The Semantic Web – ISWC 2019 : 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II*, page 146–162, Berlin, Heidelberg, 2019. Springer-Verlag.

[6] Noémie Jacquet, Vincent Guigue, Cristina E. Manfredotti, Fatiha Saïs, Stéphane Dervaux, and Paolo Viappiani. Modélisation du caractère séquentiel des repas pour améliorer la performance d'un système de recommandation alimentaire. In *Extraction et Gestion des Connaissances, EGC 2024, Dijon, France, January 22-26, 2024*, volume E-40 of *RNTI*, pages 131–142. Editions RNTI, 2024.

[7] Moise Kombolo, Jérémy Yon, François Landrieu, Brigitte Richon, Sophie Aubin, and Jean-François Hocquette. Le Thésaurus de la viande : un nouvel outil accessible à tous Une nouvelle ressource sémantique répondant aux principes de la science ouverte : le thésaurus de la viande comme outil informatique de dialogue entre les acteurs de la filière. *Viandes et Produits Carnés*, March 2022.

[8] Myriam Merad, Sophie S. Allain, Jean-Christophe Augustin, Sandrine Blanchemanche, Gilles Bornert, Michel Federighi, Michel Gautier, Guillier Laurent, Nicole Hagen-Picard, Laïla Lakhal, Eric Marchioni, Régis Pouillot, and Brigitte Roudaut. Hiérarchisation des dangers biologiques et chimiques dans le but d'optimiser la sécurité sanitaire des aliments : Méthodologie et preuve de concept. Technical Report Saisine n°2016-SA-0153, Anses, 2020.

[9] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. A survey on food computing. *ACM Computing Surveys*, 52(5) :1–36, 2019.

[10] Carlos A. Monteiro, Geoffrey Cannon, Mark Lawrence, Maria Laura da Costa Louzada, and Priscila Pereira Machado. Ultra-processed foods, diet quality, and health using the nova classification system, 2019.

[11] Dariush Mozaffarian. Dietary and policy priorities for cardiovascular disease, diabetes, and obesity : a comprehensive review. *Circulation*, 133(2) :187–225, 2016.

[12] José M. Ordovás, Lynnette R. Ferguson, Esmond S. Tai, and John C. Mathers. Personalised nutrition and health. *BMJ*, 361 :bmj.k2173, 2018.

[13] Sachit Rajbhandari and Johannes Keizer. The agrovoc concept scheme – a walkthrough. *Journal of Integrative Agriculture*, 11(5) :694–699, 2012.

[14] Mari Carmen Suárez-Figueroa. Neon methodology for building ontology networks : specification, scheduling and reuse. In *DISKI*, 2011.

[15] Christoph Trattner and David Elsweiler. Food recommender systems : important contributions, challenges and future research directions. *arXiv preprint*, arXiv :1711.02760, 2017.

[16] Ying Zhu, Xiaodong Li, and Fei Wang. Personalized nutrition recommendation algorithm based on knowledge graph. *IEEE Access*, 8 :202778–202788, 2020.

Prédiction d'événements cliniques à partir de parcours de soins représentés par des graphes de connaissances temporels

Jong Ho Jhee[1], Alberto Megina[1], Pacôme Constant Dit Beaufils[2,3], Matilde Karakachoff[3,4], Richard Redon[2], Alban Gaignard[2], Adrien Coulet[1]

[1] Inria, Inserm, Université Paris Cité, HeKA, UMR 1346, Paris, France
[2] CNRS, Inserm, Université de Nantes, Institut du Thorax, UMR 1087, Nantes, France
[3] Université de Nantes, CHU Nantes, Nantes, France
[4] Inserm, Clinique des données, CIC 1413, Nantes, France

jong-ho.jhee@inria.fr

Résumé

Contexte : *La disponibilité croissante des données de santé permet le développement de modèles capables de prédire divers événements cliniques, ce qui trouve de nombreuses applications notamment en termes d'aide à la décision clinique.* ***Méthode :*** *Nous avons créé un jeu de données synthétiques mais réalistes de patients atteints d'anévrisme intracrânien et avons considéré la tâche de prédiction de leur état en fin d'hospitalisation. La tâche de prédiction est ici ramenée à une classification d'un sous ensemble de nœuds d'un KG. Nous comparons les performances sur cette tâche en considérant comme référence des données tabulaires, puis ces mêmes données représentées sous forme de graphes. Nous avons en particulier considéré plusieurs schémas de graphes pour étudier dans quelle mesure les choix de modélisation des données individuelles d'une part, et des données temporelles d'autre part impactent les performances de la prédiction.* ***Résultats :*** *Notre étude montre que dans notre cas une représentation en graphe et des plongements appris par des réseaux convolutifs donnent les meilleures performances. Nous soulignons l'impact du type de schéma de graphe utilisé ainsi que celui de la prise en compte des litéraux. En revanche, nous nuançons l'impact relatif du choix d'encodage des données temporelles.*

Mots-clés

Graphes de connaissances temporels, plongement de graphes, Réseau convolutif de graphe, données cliniques, prédiction d'événement clinique.

Abstract

Background : *With the increasing availability of healthcare data, predictive modeling finds many applications in the biomedical domain, such as the evaluation of the level of risk for various conditions, which in turn can guide clinical decision making. However, it is unclear how knowledge graph data representations and their embedding, which are competitive in some settings, could be of interest in biomedical predictive modeling.* ***Method :*** *We simulated synthetic but realistic data of patients with intracranial aneurysm and experimented on the task of predicting their clinical outcome. We compared the performance of various classification approaches on tabular data versus a graph-based representation of the same data. Next, we investigated how the adopted schema for representing first individual data and second temporal data impacts predictive performances.* ***Results :*** *Our study illustrates that in our case, a graph representation and Graph Convolutional Network (GCN) embeddings reach the best performance for a predictive task. We emphasize the importance of the adopted schema and of the consideration of literal values in the representation of individual data. Our study also moderates the relative impact of various time encoding on GCN performance.*

Keywords

Temporal knowledge graph, Graph embedding, Graph convolutional network, Clinical data, Outcome prediction.

Cet article est la traduction de l'article disponible en anglais sur arXiv, et présenté à la conférence ESWC 2025.

1 Introduction

Les anévrismes intracrâniens sont des dilatations anormales des vaisseaux sanguins du cerveau qui présentent des risques importants pour la santé, en particulier lorsqu'ils rompent, entraînant des dommages neurologiques sévères ou la mort [18]. La capacité à prédire l'évolution et les meilleures interventions en fonction du profil des patients est cruciale pour améliorer la prise en charge des patients.
Dans ce contexte, nous visons à établir une méthode permettant d'identifier les patients présentant un risque plus élevé de complication après la rupture d'un anévrisme intracrânien. En combinant variables personnelles des patients et traitements observés lors de leur hospitalisation, nous espérons mettre en évidence des schémas cliniques et thérapeutiques permettant d'améliorer la prise en charge.

Les plongements de graphes (ou *graph embeddings*) sont des représentations des nœuds et arêtes qui composent un graphe au sein d'un espace vectoriel continu [13]. Cette transformation permet de représenter des données relationnelles complexes dans un format adapté à l'analyse computationnelle. En particulier, elle facilite l'utilisation d'algorithmes d'apprentissage automatique pour effectuer des tâches telles que la prédiction de liens, la classification de nœuds ou le clustering [28].

Dans ce travail, nous nous intéressons particulièrement aux plongements à partir de graphes de connaissances (KG), tels que définis dans le cadre du Web Sémantique [16]. Les éléments atomiques qui composent ces KG sont des triplets de la forme $\langle sujet, prédicat, objet \rangle$, où sujet et objet sont des nœuds représentant des entités, et le prédicat est une arête étiquetée et orientée, indiquant qu'une relation spécifique existe entre le sujet et l'objet [15]. Ces KG sont généralement encodés en RDF (Resource Description Framework) un standard où les entités et les prédicats sont identifiés de manière unique à l'aide d'URI (Uniform Resource Identifier), facilitant l'interopérabilité entre différents ensembles de données et ontologies.

Cependant, la mesure selon laquelle apprendre des plongements à partir de KG pourrait être utile pour la prédiction d'événements cliniques n'est pas claire [6]. Dans ce travail, nous apportons de premiers éléments de réponse en explorant trois questions scientifiques clés qui ont guidé notre travail. Premièrement, est ce que dans notre contexte une représentation en graphe est intéressante pour une tâche de prédiction. Deuxièmement, comment les approches prédictives peuvent-être affectées par le modèle choisi pour représenter des données individuelles sous forme de graphe ? Troisièmement, quel impact peut avoir les choix de modélisation pour la représentation du temps, *i.e.*, temps absolu, relatif ou une combinaison des deux, sur les performances prédictives.

Les contributions de cet article sont :

(*i*) un jeu de données synthétique mais réaliste, publiquement accessible, représentant les parcours de patients traités pour un anévrisme intracrânien rompu, ainsi que des scripts permettant de transformer ces données d'un format tabulaire vers diverses représentations sous forme de graphes ;

(*ii*) des éléments de réponse à nos trois questions, illustrant que, dans notre contexte, les plongements de graphes appris par des réseaux convolutifs de graphe surpassent les autres approches, qu'une représentation plus compacte des variables des patients est associée à de meilleures performances, et que la représentation du temps n'a pas d'impact significatif sur les performances prédictives.

La suite de cet article présente les travaux connexes dans la section 2.1, les matériel et méthodes en section 3, les résultats empiriques et leur interprétation en section 4, et enfin une discussion en section 5.

2 Etat de l'art

2.1 Schéma pour les données cliniques

Plusieurs schémas de données ont été proposés pour modéliser et faciliter l'échange de données cliniques. Par exemple, FHIR [22] et OMOP CDM [27] sont des standards visant à résoudre les problèmes d'interopérabilité des données cliniques. Des transformations RDF de FHIR [23] et du OMOP CDM ont été proposées, cependant aucune d'elles n'est largement adoptée [4, 29].
Par ailleurs, deux ontologies récentes ont été proposées pour représenter les données cliniques individuelles sous forme de KG.
La première est SPHN (Swiss Personalized Health Network) [26], qui a été adoptée par les hôpitaux universitaires suisses afin d'améliorer le partage et l'intégration des données. Comme illustré par la Figure 1, dans ce modèle le patient est une entité centrale pouvant être associée à des diagnostics, des administrations de médicaments et des procédures, chacune pouvant être liée à des données temporelles sous forme de littéraux RDF.
La seconde est l'ontologie CARE-SM (Care and Registry Semantic Model) [17]. Elle a été conçue pour représenter les données cliniques dans le contexte des maladies rares et repose largement sur la réutilisation de l'ontologie SIO (Semantic science Integrated Ontology) [9]. Comme illustré Figure 2, l'originalité de CARE-SM réside dans son utilisation des RDF quads, qui permettent d'associer un contexte à chaque élément de donnée via un *graphe nommé RDF*. De plus, ces graphes nommés peuvent être utilisés pour représenter des lignes temporelles d'événements cliniques.
Dans ce travail, nous nous concentrons sur SPHN et CARE-SM pour trois raisons principales. Premièrement, ces ontologies fournissent des spécifications suffisamment précises pour permettre la représentation d'un jeu de données cliniques quelconque, ce qui n'est pas le cas de FHIR-RDF. Deuxièmement, elles proposent des choix de modélisation très différents. Troisièmement, elles sont adoptées dans le cadre de projets à grande échelle.

2.2 Le temps dans les KG

OWL-Time est une ontologie standard qui fournit classes et prédicats pour représenter le temps, la durée et les intervalles [5]. En particulier, elle permet d'instancier des relations entre événements en utilisant l'algèbre des intervalles d'Allen et peut être associée à des mécanismes de raisonnement temporel [30].
SPHN et CARE-SM offrent tous deux plusieurs façons de représenter le temps, y compris par l'utilisation d'OWL-Time. En pratique, on peut choisir de se restreindre au temps absolu en associant uniquement des horodatages aux événements, mais il est aussi possible d'utiliser le temps relatif avec des relations de précédence entre les événements, ou encore d'utiliser à la fois temps absolu et relatif.
De plus, pour une représentation du temps relatif, il faut décider d'un *niveau de saturation* des relations temporelles. Ce niveau peut aller d'une simple séquence, où seuls les événements directement successifs sont liés, à un niveau

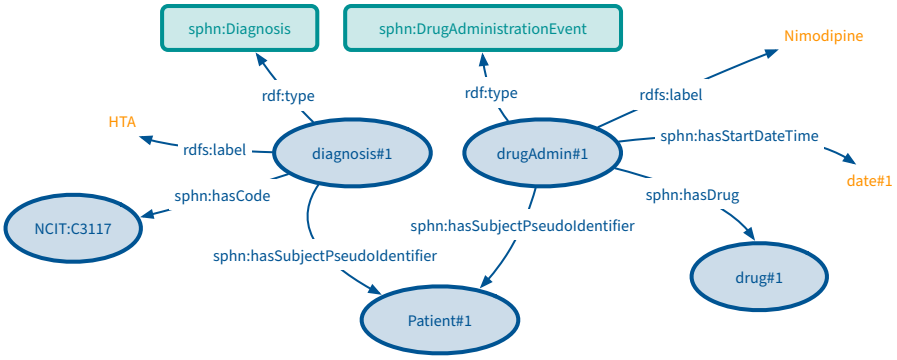

FIGURE 1 – Exemple de données cliniques individuelles représentées avec le schéma SPHN. Les informations temporelles sont représentées par des littéraux RDF associés aux événements via la propriété `sphn:hasStartDateTime`.

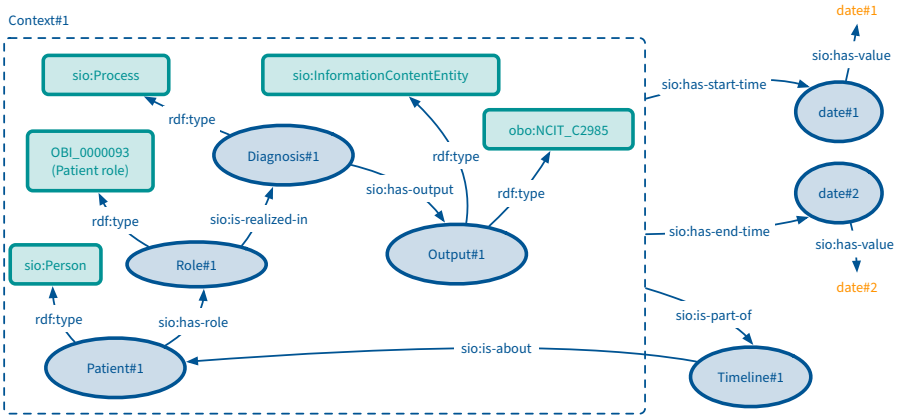

FIGURE 2 – Exemple de données cliniques individuelles représentées avec le schéma CARE-SM. L'information temporelle est directement liée au graphe nommé `Context#1` avec des dates de début et fin. Plusieurs événements peuvent être associés à une séquence temporelle d'événements via la propriété `sio:is-part-of`.

pleinement saturé où chaque événement est lié temporellement à tous les autres. Bien qu'il soit évident qu'un graphe pleinement saturé présente plusieurs inconvénients (*e.g.*, la connectivité élevée des nœuds rend l'exploration du graphe complexe), le choix d'une modélisation temporelle adaptée à une tâche spécifique n'est pas toujours évident [30].

2.3 Plongements et classification de nœuds

De nombreuses approches existent pour représenter des KG dans un espace latent, et une catégorisation de celles-ci a été proposée dans [3]. À titre d'illustration, TransE [1] est une approche de plongement qui fonctionne au niveau des triplets en minimisant la fonction de score suivante :

$$f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_1,$$

où $\mathbf{h}$, $\mathbf{r}$ et $\mathbf{t}$ sont respectivement les vecteurs de plongement associés à l'entité de tête, de la relation et de l'entité de queue. Ici, la relation entre la tête et la queue (*i.e.*, le sujet et l'objet) peut être vue comme une translation r dans l'espace de plongement.

RDF2Vec suit une approche très différente, opérant au niveau des séquences [24]. Il utilise des marches aléatoires extraites du KG. Ces séquences de nœuds, d'arêtes ou de sous-arbres servent à alimenter un modèle word2vec qui génère des plongement pour chaque nœud en maximisant la probabilité du nœud suivant, étant donné une séquence de nœud. Par exemple, le modèle *Continuous Bag-of-Words*, l'un des algorithmes associés à word2vec, maximise la probabilité moyenne logarithmique du nœud cible, étant donnée une séquence de nœuds :

$$\frac{1}{T}\sum_{t=1}^{T} \log p(e_t | e_{t-c}, ..., e_{t+c}),$$

où e_t est un nœud cible et c désigne la fenêtre de contexte. Contrairement à TransE et RDF2Vec, qui fonctionnent respectivement au niveau des triplets et des séquences, les GCNs (*Graph Convolutional Networks*) [20] opèrent au niveau du voisinage des nœuds. Introduits pour la classification de graphes, ils ont été étendus à la classification de nœuds et à la prédiction de liens dans les KG [25]. Les GCNs calculent les plongements d'un nœud en prenant en compte son voisinage dans le graphe. Ils peuvent être vus comme une approche de transmission de messages à travers plusieurs couches, où le plongement $h_i^{(l+1)}$ d'un nœud i à la couche $(l+1)$ dépend des plongements de ses voisins à

la couche (l), selon :

$$h_i^{(l+1)} = \sigma\left(\underbrace{\sum_{j \in \mathcal{N}_i} \frac{1}{c_i} W^{(l)} h_j^{(l)}}_{\text{Voisinage}} + \underbrace{W^{(l)} h_i^{(l)}}_{\text{Auto-connexion}} \right).$$

La convolution sur les nœuds voisins j de i est calculée avec une matrice de poids $W^{(l)}$ et normalisée par une constante c_i. Le dernier terme assure l'auto-connexion, c'est-à-dire que le plongement du nœud i à la couche $(l+1)$ dépend aussi de son plongement à la couche (l).
Les *Relational Graph Convolutional Networks* (RGCN) [25] sont une variante des GCNs prenant en compte les données multi-relationnelles, c'est-à-dire qu'elles différencient un même voisin selon le type de relation qui le relie à i. Ceci est particulièrement adapté aux KG du web sémantique, qui utilisent divers prédicats pour représenter des relations associées à des sémantiques différentes. Pour cela, les RGCNs intègrent les plongements d'entités avec des relations multiples dans leur agrégation des plongements du voisinage suivant :

$$h_i^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)} \right)$$

où la convolution sur les nœuds voisins est réalisée avec une matrice de poids spécifique $W_r^{(l)}$ pour chaque type de relation $r \in \mathcal{R}$, et $c_{i,r}$ est une constante de normalisation correspondant au nombre de voisins $|\mathcal{N}_i^r|$.
Dans ce travail, nous considérons TransE et RDF2Vec comme deux approches de référence pour le plongement des KG (KGE) et utilisons RGCN comme candidat adapté pour prendre en compte la diversité des types de relations.
Les valeurs associées aux entités sous forme de littéraux sont généralement ignorées par les approches KGE. Toutefois, plusieurs travaux ont étudié l'intérêt de leur considération. Par exemple, *LiteralE* [21] propose une approche où un nœud est représenté par deux vecteurs : le premier pour le plongement du nœud lui-même, et le second pour les littéraux numériques qui lui sont associés. Ces deux vecteurs sont ensuite combinés avant d'être intégrés à une fonction de score. *KEN* [7] utilise un encodeur basé sur un réseau de neurones unique pour injecter les valeurs littérales dans le même espace vectoriel que les entités. D'autres approches ont été proposées pour traiter les littéraux textuels ou combiner plusieurs types de littéraux [12].
Dans ce travail, nous explorons trois approches de plongement, mais nous nous concentrons sur une seule tâche : la classification de nœuds. Cette tâche d'apprentissage consiste à estimer la probabilité qu'une entité, non explicitement définie dans un KG, appartienne à un type donné [28]. L'hypothèse sous-jacente des KGE est que l'espace de plongement capture efficacement les variables et la structure du graphe original. En pratique, la configuration spatiale des vecteurs doit refléter l'information relationnelle observée dans le KG, permettant ainsi de prédire la classe des nœuds. Cette tâche est formalisée dans la section *Matériels et Méthodes* de cet article.

2.4 Plongements des KG temporels

Un graphe de connaissances temporel (*Temporal Knowledge Graph*, TKG) étend un KG existant en y intégrant des informations temporelles. Cette extension pose des défis majeurs, car elle nécessite d'intégrer la temporalité associée aux entités et leurs relations dans les modèles afin de capturer avec précision la dynamique de celles-ci [8]. Une modélisation efficace de l'aspect temporel est essentielle pour les applications où l'évolution des relations est déterminante, comme dans les données cliniques où le moment des traitements, interventions peuvent avoir une influence critique sur les résultats.
Les modèles de plongement des TGK utilisent des triplets marqués par un horodatage, formant ainsi des quadruplets. L'apprentissage dans ce contexte est alors considéré comme la représentation de chaque instantané temporel du graphe. Toutefois, l'intervalle temporel peut être une information clé, ou bien le graphe peut ne pas être compatible avec le format des quadruplets, car tous les triplets ne possèdent pas nécessairement d'information temporelle. L'encodage ou la transformation des informations temporelles sous forme de vecteurs de littéraux ou de relations peut alors être appliqué [2].

3 Matériels et Méthodes

3.1 Génération de données synthétiques

Nous avons construit un jeu de données synthétiques de 10 000 patients, 30 variables cliniques et une variable réponse. Parmi ces 30 variables, 8 sont associées à une information temporelle et sont alors désignées par le terme d'événement, afin de les distinguer des variables démographiques ou historiques qui ne sont pas associés à un temps.
Dans ce jeu de données, les événements ont la particularité d'être associés à un information temporelle unique, qui correspond au temps écoulé entre l'admission à l'hôpital et la première occurrence de cet événement durant le séjour.
Pour rendre notre jeu de données synthétiques aussi réaliste que possible, les variables des patients et les événements ont été générés en fonction d'observations faites sur un jeu de données réel de 552 patients diagnostiqués avec un anévrisme intracrânien rompu, fourni par le Centre Hospitalier Universitaire de Nantes. L'accès à ce jeu de données a été accordé par le comité d'éthique local.
Dans un premier temps, nous avons estimé la distribution la plus proche de chaque variable du jeu de données réel à l'aide d'un test de Kolmogorov–Smirnov. Par exemple, nous avons observé que la durée du séjour à l'hôpital suivait une loi d'extremum généralisée. Ensuite, après avoir effectué une factorisation pour les variables catégorielles, nous avons calculé les corrélations pour chaque paire de variables et les probabilités de transition entre les types d'événements. Nous avons observé que la durée du séjour à l'hôpital était fortement corrélée avec le nombre de procédures médicales reçues, et qu'il y avait une probabilité élevée que le Paracétamol soit administré après la Nimodipine. La Figure 3 montre un diagramme de *Sankey* construit à partir des probabilités de transition observées, qui résume visuel-

lement les parcours de soins possibles. Pour simplifier la représentation du temps dans notre jeu de données, les 8 événements ont été binarisés par paire, avec une valeur à un si le patient a observé une transition entre le premier et le deuxième événement, et à zéro sinon. Après validation par un clinicien de la cohérence des distributions, corrélations et probabilités de transition, nous les avons utilisées pour contraindre la génération de nos données synthétique.

Enfin, nous avons généré la variable réponse avec l'une des trois valeurs distinctes {*BackHome, Rééducation, Décès*} respectivement associées aux proportions suivantes dans les données synthétique : 44,14%, 43,33% et 12,53%, pour refléter la distribution du jeu de données réel.

3.2 Représentation en graphe des données

Nous avons développé des transformations de données tabulaire en utilisant différents choix de modélisation. En utilisant des patrons RDF et des règles, nous avons généré des graphes instanciant l'ontologie SPHN :

- `SPHN-nl` (sans littéraux) où tous les littéraux ont été supprimés ;
- `SPHN-nt` (sans temps) où l'information temporelle a été supprimée ;
- `SPHN-ts` (*timestamps*) avec uniquement des temps absolus ;
- `SPHN-tr` (relations temporelles) avec un prédicat `time:before` entre événements directement successifs ;
- `SPHN-tsr` (*timestamps* et relations) avec *timestamps* et des relations entre événements successifs ;
- `SPHN-sat1` (niveau de saturation 1) où `SPHN-tr` est enrichi avec des prédicats `time:before` en appliquant une fois la règle `time:before` ∘ `time:before` ⊑ `time:before` ;
- `SPHN-sat2` (niveau de saturation 2) en appliquant la même transitivité une seconde fois.

La Figure 4 illustre les *timestamps* et les relations temporelles entre les événements, à trois niveaux de saturation correspondant à `SPHN-tr`, `SPHN-sat1` et `SPHN-sat2`.

En utilisant des modèles RDF, nous avons également généré un graphe qui instancie l'ontologie CARE-SM. Dans ce cas, nous avons effectué une étape supplémentaire où les quadruplets ont été transformés en triplets, en particulier en utilisant un lien `nvasc:hasTimePoint` pour associer directement des instances telles que les diagnostics ou les administrations de médicaments à leur temps absolu. Cela est motivé par le fait que les approches de KGE ne sont pas conçues pour fonctionner avec des quads, mais uniquement des triplets. Le graphe résultant est nommé *CARE-SM**. Nous avons généré les mêmes variantes de graphes que pour SPHN, mais nous ne présentons ici que la version `CARE-SM*-ts` (*timestamps* seulement).

Nous effectuons deux dernières transformations sur chaque variante de nos graphes : premièrement, pour nous assurer que l'orientation des prédicats n'influence pas nos expériences, des relations inverses sont systématiquement ajoutées ; nous avons encodé les littéraux des *timestamps* sous forme de nombres continus sur [0, 1] en utilisant une transformation par quantile. Cela permet d'étaler les valeurs les plus fréquentes et de réduire l'impact des valeurs aberrantes [10]. Les scripts de ces transformations sont disponibles à `https://github.com/TeamHeka/neurovasc`.

3.3 Plongements de KG et prédiction

L'objectif de notre tâche de prédiction est de prévoir l'évolution clinique (ou réponse) des patients en fonction de leurs variables et des événements. Dans le cas des données sous forme de graphes, nous modélisons cette tâche comme un problème de classification de nœuds visant à les associer à la classe correspondant à leur réponse.

3.3.1 RGCN pour la prédiction de l'évolution clinique

Dans cette section, nous décrivons en détail le modèle de KGE, *RGCN+Literals*, pour la prédiction de l'évolution clinique. L'architecture générale est illustrée en Figure 5. Soit un graphe $G = (\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{X})$, où $\mathcal{V}$ désigne l'ensemble des nœuds (entités), $\mathcal{E}$ l'ensemble des arêtes, $\mathcal{R}$ l'ensemble des relations (prédicats), et $\mathcal{X} \in \mathbb{R}^{n \times d_0}$ désigne les plongements de dimension d_0. La représentation d'un nœud cible (patient) $h_i^{(1)} \in \mathbb{R}^{d_1}$ pour $i \in |\mathcal{V}|$ après une première couche est définie par :

$$h_i^{(1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(0)} x_j + W_0^{(0)} x_i \right),$$

où $x_i \in \mathcal{X}$ est le plongement initial du patient, $W_r \in \mathbb{R}^{d_o \times d_1}$ désigne une matrice de poids pour chaque relation r et σ est une fonction d'activation non linéaire. De plus, le nombre de paramètres augmente avec le nombre de relations, ce qui peut entraîner un sur-apprentissage pour certaines relations rares. Pour cette raison, nous utilisons une décomposition pour la régularisation du modèle [25].

Le plongement final du nœud patient est extrait après avoir passé L couches empilées de RGCN, et la fonction *softmax* est appliquée pour générer la probabilité résultat. Le modèle est entraîné en minimisant la fonction de perte d'entropie croisée sur les nœuds de patients :

$$\mathcal{L} = -\sum_{p \in \mathcal{P}} \sum_{k=1}^{K} y_{pk} \log z_{pk},$$

où $\mathcal{P}$ est l'ensemble des nœuds de patients dans l'ensemble d'entraînement, K le nombre de classes et z_{pk} la probabilité de la réponse.

3.3.2 RGCN avec littéraux

Dans notre KG clinique, certaines variables sont représentées sous la forme de littéraux (*e.g.*, l'âge d'un patient). Cependant, le modèle RGCN ne prend en compte que les entité et relations, et ne considère donc pas les littéraux. Pour faire face à ce problème, nous proposons un modèle appelé RGCN+Literals (RGCN+lit) qui utilise une fonction supplémentaire pour les littéraux. Avant l'entrée des plongements initiaux dans le RGCN, une fonction de perceptron multi-couches (MLP) est utilisée pour transformer la valeur du littéral en vecteur :

$$x_{literal} = \sigma(Wv + b),$$

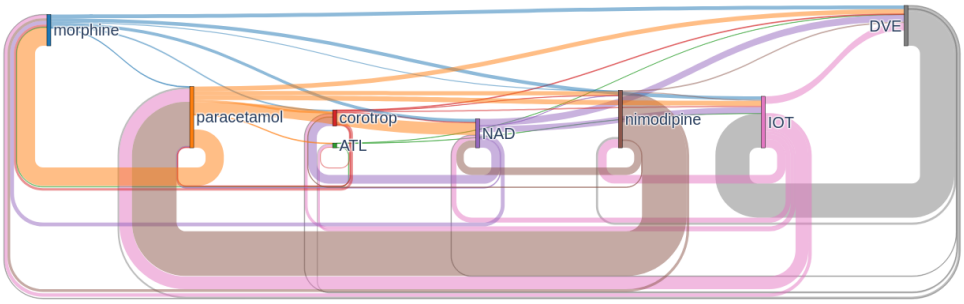

FIGURE 3 – Représentation visuelle des parcours de soins où les connexions plus larges entre les événements de soins correspondent à des probabilités de transition plus élevées. `morphine` : utilisation de morphine, `paracetamol` : utilisation de paracétamol, `corotrop` : utilisation de milrinone, `ATL` : angioplastie transluminale percutanée, `NAD` : utilisation de noradrénaline, `nimodipine` : utilisation de nimodipine, `IOT` : intubation orotrachéale, et `DVE` : drainage ventriculaire externe.

où v est la valeur de l'attribut, σ la fonction d'activation non linéaire, W la matrice de poids et b le biais. Des fonctions d'encodage pourraient remplacer le MLP.

4 Résultats expérimentaux

A partir de notre jeu de données synthétiques, nous comparons diverses approches de prédiction de la réponse à travers trois expériences distinctes. Chacune d'elle est répétée dix fois, pour chaque modèle, afin d'évaluer la variabilité des performances. À chaque fois, les 10 000 patients sont répartis aléatoirement en ensembles d'entraînement (80%), de validation (10%) et de test (10%). Données et programmes sont disponibles sur `https://anonymous.4open.science/r/Predicting-clinical-outcomes-with-TKG-B2B9`.
Les trois expériences visent à comparer :

- **Données tabulaires *vs.* graphe.** Nous comparons la performance des approches prédictives standard appliquées sur des données tabulaires avec celles des diverses approches de plongement de KG pour évaluer l'intérêt de l'information multi-relationnelle pour la tâche de prédiction.
- **Ontologie SPHN *vs.* CARE-SM.** Nous comparons l'impact sur la prédiction de l'utilisation de l'une ou l'autre des structures pour évaluer si la structure du graphe affecte la performance.
- **Modélisations temporelles variées.** Nous comparons l'impact sur la prédiction des choix de modélisations temporelles, tels que l'absence de données temporelles, uniquement le temps absolu, uniquement le temps relatif, les deux à la fois, et de différents niveaux de saturation.

4.1 Données tabulaires *vs.* graphe

Nous avons considéré les méthodes suivantes pour évaluer la performance des méthodes portant sur des données tabulaires (*baseline*) : régression logistique (LR), forêts aléatoires (RF), réseau de neurones (NN) pour les comparer aux approches KGE. La méthode RF a été configurée avec 100 arbres, et la méthode NN avec trois couches avec des dimensions cachées de [100, 50, 10] et une fonction d'activation tangente hyperbolique.

Pour cette première comparaison avec diverses approches KGE, nous avons choisi de ne présenter que ceux avec l'ontologie SPHN car ce sont les meilleurs. De plus, afin d'assurer une comparaison équitable avec les données tabulaires (qui n'encodent que la séquence d'événements), le graphe `SPHN-tr` a été considéré comme le graphe SPHN de référence. Pour représenter les trois principales familles de KGE, nous avons considéré TransE, RDF2Vec et RGCN+lit. Tous les modèles ont été configurés avec une dimension de vecteurs d'entrée de 100. Pour TransE, la norme L1 (1-norm) est appliquée pour la régularisation de la fonction de score. Pour RDF2Vec, dix marches aléatoires d'une profondeur maximale de trois pour chaque nœud ont été appliquées. Les représentations des patients obtenues à partir des deux premières approches KGE sont fournies en entrée d'un modèle NN en tant que classificateur.

Pour RGCN+lit, trois couches RGCN sont appliquées afin d'agréger l'information du voisinage à distance 3 des nœuds patients, et la fonction d'activation non linéaire utilisée est la *Parametric Rectified Linear Unit* (PReLU) [14]. Tous les modèles ont été optimisés à l'aide de l'optimiseur Adam [19] avec un taux d'apprentissage de 1e-3 et une pénalisation des poids (*weight decay*) de 5e-4.

Les performances obtenues sont présentées dans le Tableau 1, notamment à l'aide du F1-score, c'est à dire la moyenne harmonique entre la précision et le rappel, et l'aire sous la courbe ROC (noté AUC). Les trois approches de référence basées sur les données tabulaires ont donné de faibles performances (F1-score = [0.44, 0.49], AUC = [0.63, 0.71]). RF a montré les meilleures performances (AUC = 0.71), suivi de près par LR. Ces deux modèles ont affiché un F1-score relativement bon pour l'évolution clinique *BackHome*.

Pour les données de type graphe, ni TransE ni RDF2Vec ne semblent parvenir à prédire correctement les réponses (AUC = [0.49, 0.5]). Cependant, le modèle RGCN+lit ob-

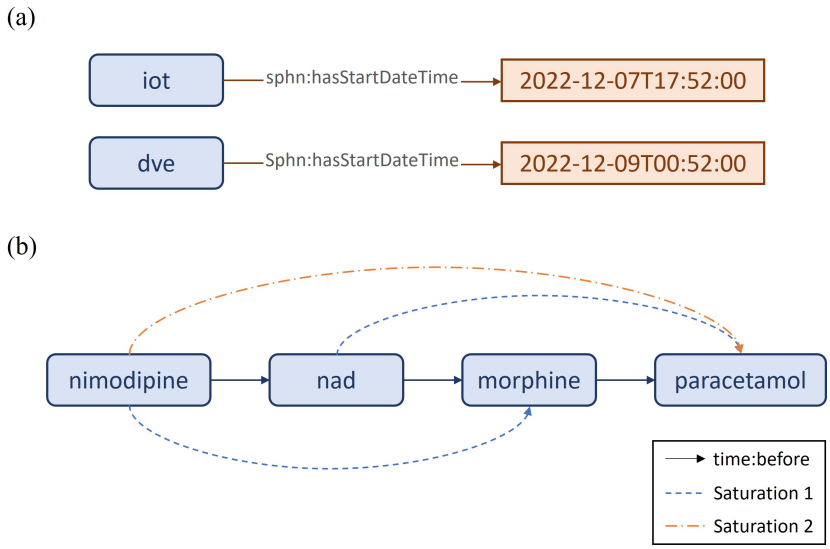

FIGURE 4 – Exemples d'information temporelle : (a) deux événements associés à un *timestamp* ; (b) séquence d'événements liés par des relations `time:before`. Les lignes continues représentent les relations entre événements directement successifs. L'application d'une règle de transitivité une fois ajoute 2 relations en ligne pointillée (saturation 1) et deux fois, ajoute la dernière relation représentée par la ligne pointillée (saturation 2). `iot` : intubation orotrachéale, `dve` : drainage ventriculaire externe, `nad` : nicotinamide adénine dinucléotide.

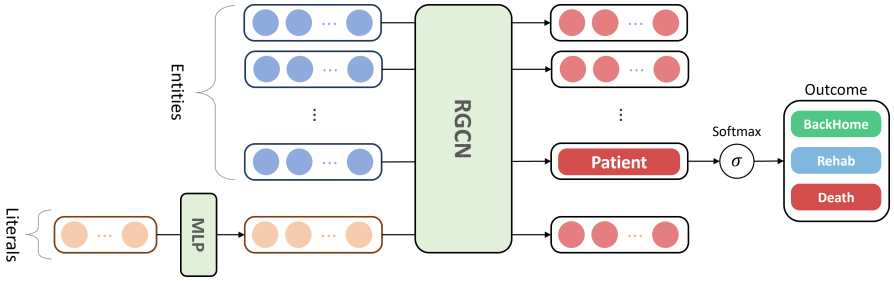

FIGURE 5 – Une illustration du modèle appelé RGCN+lit pour la prédiction de l'évolution clinique.

tient les meilleurs résultats (F1 = 0.78, AUC = 0.91).

4.2 Ontologies SPHN *vs.* CARE-SM

Comme dans la première expérience, nous considérons TransE, RDF2Vec et RGCN+lit, mais appliqués ici à un KG instanciant soit l'ontologie SPHN, soit CARE-SM. La configuration des modèles et les hyperparamètres sont identiques à ceux de la première expérience. Comme CARE-SM utilise un nombre plus important de propriétés et donc des chemins plus longs pour connecter les patients à leurs variables, nous avons mené une expérience supplémentaire sur le KG CARE-SM avec un RGCN constitué de cinq couches. Cela vise à vérifier si considérer des voisins à une distance de 5 sauts permet de capturer une information suffisante pour prédire la réponse. Les performances obtenues sont rapportées dans le Tableau 2. TransE et RDF2Vec ont montré des performances relativement faibles sur les deux KG. Le RGCN sur SPHN a montré la meilleure performance (AUC=0.91). Pour CARE-SM, tous les modèles ont des difficultés à prédire la réponse. En utilisant cette ontologie, nous notons que le RGCN avec trois couches n'est pas plus performant que TransE ou RDF2Vec (AUC=0.50). Augmenter le nombre de couches à 5 n'améliore pas les performances (AUC=0.50). Nous notons qu'aucun modèle ne parvient réellement à prédire la réponse *Décès* sauf SPHN-ts avec RGCN3+lit.

4.3 Modélisation des informations temporelles

Dans cette expérience, nous comparons les performances prédictives des graphes associés à différentes modélisations du temps comme indiqué dans la Section 3.2. Toutes les expériences sont menées en utilisant le modèle RGCN+lit, sauf dans le cas où nous ne considérons pas les littéraux et utilisons RGCN. Sans littéraux, le modèle n'est pas performant, comme avec les données tabulaires. L'AUC augmente d'environ 33% en moyenne lorsque les littéraux sont ajoutés. Cette augmentation provient principalement d'une meilleure prédiction de la réponse *Décès*. Lorsque les informations temporelles (*timestamps*) sont ajoutées à

TABLE 1 – Comparaison de la prédiction des réponses sur les données tabulaires et sur le graphe (SPHN-tr). RGCN3+lit fait référence à RGCN+lit avec 3 couches.

Type	Modèle	F1-score					Précision	AUC
		BackHome	Rééducation	Décès	Macro	Pondéré		
Tabulaire	LR	0.63±0.02	0.55±0.02	0.25±0.05	0.47±0.03	0.55±0.02	0.56±0.02	0.70±0.02
	RF	0.63±0.01	0.55±0.02	0.28±0.04	0.49±0.02	0.55±0.01	0.56±0.01	0.71±0.01
	NN	0.58±0.03	0.48±0.03	0.26±0.04	0.44±0.02	0.50±0.02	0.50±0.02	0.63±0.02
Graphe (SPHN-tr)	TransE	0.49±0.04	0.40±0.10	0.02±0.04	0.30±0.03	0.40±0.03	0.43±0.02	0.50±0.01
	RDF2Vec	0.50±0.05	0.39±0.14	0.01±0.02	0.30±0.03	0.39±0.04	0.44±0.02	0.49±0.01
	RGCN3+lit	**0.84±0.01**	**0.76±0.02**	**0.64±0.08**	**0.75±0.03**	**0.75±0.02**	**0.78±0.01**	**0.91±0.01**

TABLE 2 – Comparaison des performances de SPHN (SPHN-ts) et CARE-SM (CARESM*-ts). RGCN3+lit et RGCN5+lit font référence à RGCN+lit avec respectivement 3 et 5 couches. CARESM* est la variante de CARE-SM. Voir Section 3.2.

KG	Modèle	F1-score					Précision	AUC
		BackHome	Rééducation	Décès	Macro	Pondéré		
SPHN-ts	TransE	0.51±0.07	0.33±0.16	0.02±0.04	0.29±0.04	0.37±0.05	0.43±0.02	0.50±0.01
	RDF2Vec	0.49±0.04	0.42±0.09	0.01±0.03	0.30±0.02	0.40±0.02	0.44±0.01	0.50±0.02
	RGCN3+lit	**0.83±0.02**	**0.76±0.02**	**0.66±0.08**	**0.75±0.03**	**0.78±0.02**	**0.78±0.02**	**0.91±0.01**
CARESM*-ts	TransE	0.47±0.04	0.44±0.04	0.02±0.03	0.31±0.01	0.40±0.01	0.43±0.01	0.49±0.01
	RDF2Vec	0.51±0.07	0.38±0.11	0.00±0.00	0.29±0.02	0.39±0.03	0.44±0.02	0.50±0.01
	RGCN3+lit	0.53±0.08	0.30±0.17	0.00±0.00	0.28±0.04	0.37±0.05	0.44±0.01	0.50±0.02
	RGCN5+lit	0.48±0.08	0.30±0.19	0.00±0.00	0.26±0.04	0.34±0.05	0.44±0.05	0.50±0.01

SPHN-nt, l'AUC augmente en moyenne de 7%. L'ajout des relations temporelles augmente également l'AUC de 7%. Ajouter les deux informations temporelles montre également des résultats similaires. Avec la saturation, nous observons une légère amélioration des performances pour le résultat *Décès*, bien que la performance globale soit similaire à SPHN-tsr. Finalement, l'ajout d'informations temporelles améliore la prédiction du modèle, mais le type de modélisation temporelle et le niveau de saturation n'apporte pas de différence significative.

5 Discussion

Premièrement, l'analyse comparative révèle qu'en termes de précision et de F1-score, le modèle RGCN+lit surpasse les méthodes de base sur les données tabulaires, ou d'autres approches KGE. Nous pensons que cette différence s'explique par le fait que le RGCN permet au modèle d'agréger les informations provenant de plusieurs voisins à plusieurs sauts du nœud patient, ce qui ne peut pas être réalisé avec une modélisation relationnelle unique sur les données tabulaires, ou avec TransE ou RDF2Vec qui ne considèrent que partiellement les voisins distants de plusieurs sauts. Pour mieux considérer l'aspect séquentiel des données tabulaires, une étude complémentaire pourrait considérer des réseaux neuronaux récurrents.

Deuxièmement, nous avons observé que le choix du schéma impacte la performance prédictive. En particulier, dans notre configuration, SPHN, schéma plus compact et orienté patient, est plus favorable que CARE-SM pour les tâches de prédiction. Cela pourrait s'expliquer par la plus grande distance entre le patient et ses variables cliniques dans CARE-SM. Cependant, augmenter la taille du voisinage à cinq saut n'a pas résolu le problème. Nous notons que nous n'avons pas étendu davantage le nombre de sauts en raison du coût computationnel associé. Une étude plus complète pourrait être menée en intégrant d'autres ontologies que SPHN et CARE-SM, comme FHIR-RDF ou Phenopackets.

Troisièmement, l'ajout d'information temporelle aide RGCN+lit à classer correctement les nœuds patients, mais nous n'avons pas observé de différence de performance associée aux différents choix de modélisation du temps. Nous reconnaissons que nous nous sommes concentrés sur des modélisations temporelles assez simples et que des scénarios plus complexes existent, comme ceux associés aux graphes dynamiques par exemple.

Dans l'ensemble, notre étude montre que les modèles KGE tels que RGCN+lit sont une approche prometteuse pour la modélisation prédictive dans le domaine de la santé. Cependant, nous notons un fort déséquilibre de classes dans notre

TABLE 3 – Les performances de RGCN+lit sur SPHN avec différentes informations temporelles et modélisation. RGCN sans littéraux est appliqué à `SPHN-nl`.

KG	F1-score					Précision	AUC
	BackHome	Rééducation	Décès	Macro	Pondéré		
SPHN-nl	0.64±0.03	0.46±0.11	0.05±0.07	0.38±0.06	0.49±0.06	0.53±0.04	0.64±0.06
SPHN-nt	0.75±0.02	0.65±0.02	0.55±0.06	0.65±0.02	0.68±0.01	0.68±0.01	0.85±0.01
SPHN-ts	0.83±0.02	**0.76±0.02**	0.66±0.08	0.75±0.03	0.78±0.02	0.78±0.02	**0.91±0.01**
SPHN-tr	**0.84±0.01**	**0.76±0.02**	0.64±0.08	0.75±0.03	0.75±0.02	**0.78±0.01**	**0.91±0.01**
SPHN-tsr	0.83±0.02	**0.76±0.02**	0.66±0.04	0.75±0.02	**0.78±0.01**	**0.78±0.01**	**0.91±0.01**
SPHN-sat1	0.83±0.01	**0.76±0.02**	0.64±0.06	0.75±0.02	**0.78±0.01**	**0.78±0.01**	**0.91±0.01**
SPHN-sat2	0.83±0.01	**0.76±0.02**	**0.68±0.05**	**0.76±0.02**	0.78±0.02	0.78±0.02	**0.91±0.01**

jeu de données (44,14%, 43,33% et 12,53% pour *BackHome*, *Rehabilitation*, *Décès*, respectivement), ce qui reflète la réalité du terrain. Cela pourrait expliquer en partie la difficulté rencontrée par la plupart des approches à prédire la classe *Décès*. Cela pourrait être atténué par des techniques de sur-échantillonnage. De plus, notre étude considère uniquement la tâche de classification des nœuds, tandis que la tâche aurait pu être modélisée comme une prédiction de lien ou de classification de triplets. D'autres études seraient nécessaires pour évaluer si nos conclusions restent valables dans le contexte d'autres tâches d'apprentissage.

De plus, nous avons observé qu'inclure des représentations vectorielles de littéraux dans le modèle améliore les performances. Dans cette étude, notre modèle se concentre sur l'incorporation simple des littéraux numériques, y compris des informations temporelles. Cependant, nous prévoyons de développer un modèle capable de traiter des littéraux multimodaux, tels qu'une combinaison de textes et de littéraux numériques [12].

L'évaluation efficace des modèles de plongement de KG est cruciale pour faire progresser le domaine et garantir que les modèles développés sont robustes, précis et utiles dans des applications pratiques [11]. Cependant, elle souffre du manque de models standardisés, notamment pour la dimension temporelle. Dans cette étude, nous avons particulièrement cherché à faire avancer cet agenda en proposant une tâche du monde réel, un jeu de données partagé et des expériences de référence documentées, qui pourront servir de base pour la modélisation prédictive à partir de graphes.

Un domaine important pour les recherches futures est d'évaluer l'impact des variables individuelles des patients sur les résultats des prédictions. En analysant l'importance relative des différentes variables, telles que les conditions médicales spécifiques ou les facteurs démographiques, les chercheurs peuvent affiner leurs modèles pour se concentrer sur les attributs les plus prédictifs. Cette approche ciblée peut améliorer l'efficacité du modèle et garantir que les informations cliniques les plus pertinentes sont prioritaires dans la prise de décision.

Des efforts sont en cours pour obtenir un ensemble de données réelles de patients afin de tester ces modèles et générer des prédictions basées sur celles-ci en lien avec le CHU de Nantes. A long terme, l'objectif est d'atteindre une validation clinique des modèles prédictifs. Cette étape implique d'appliquer les modèles aux données réelles des patients et d'évaluer rigoureusement leur performance dans un contexte clinique. La validation clinique est essentielle pour garantir que les modèles soient à la fois théoriquement solides et pratiquement utiles pour améliorer les résultats des patients. Cette phase de recherche nécessitera une collaboration étroite avec les soignants et les institutions pour tester les modèles dans des environnements réels et recueillir des retours pour un affinement ultérieur.

En conclusion, notre étude démontre dans le cadre de notre cas d'utilisation le potentiel des plongements de graphes de connaissances pour prédire l'évolution clinique des patients. Un travail conséquent reste à accomplir. Les recherches futures pourront s'appuyer sur ces premières bases pour développer des outils prédictifs précis et cliniquement pertinents, en continuant à affiner les modèles, à améliorer le réalisme des données synthétiques, à évaluer sur données réelles et à ouvrir ces résultats à d'autres problématiques cliniques.

Références

[1] A. BORDES et al. "Translating Embeddings for Modeling Multi-relational Data". In : *Advances in Neural Information Processing Systems*. T. 26. 2013.

[2] B. CAI et al. "Temporal knowledge graph completion : a survey". In : *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 2023, p. 6545-6553.

[3] H. CAI, V. W. ZHENG et K. C.-C. CHANG. "A comprehensive survey of graph embedding : Problems, techniques, and applications". In : *IEEE transactions*

on knowledge and data engineering 30.9 (2018), p. 1616-1637.

[4] A. CHYTAS, N. BASSILEIADES et P. NATSIAVAS. "Mapping OMOP-CDM to RDF : Bringing Real-World-Data to the Semantic Web Realm". In : *Digital Health and Informatics Innovations for Sustainable Health Care Systems*. 2024, p. 1406-1410.

[5] S. J. D. COX et al. *Time Ontology in OWL*. W3C Candidate Recommendation. Accessed : 2024-12-18. Nov. 2022.

[6] H. CUI et al. "A Survey on Knowledge Graphs for Healthcare : Resources, Application Progress, and Promise". In : *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*. 2023.

[7] A. CVETKOV-ILIEV, A. ALLAUZEN et G. VAROQUAUX. "Relational data embeddings for feature enrichment with background information". In : *Machine Learning* 112.2 (2023), p. 687-720.

[8] L. DALL'AMICO, A. BARRAT et C. CATTUTO. "An embedding-based distance for temporal graphs". In : *Nature Communications* 15.1 (2024), p. 9954.

[9] M. DUMONTIER et al. "The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery". In : *Journal of Biomedical Semantics* 5 (2014).

[10] W. EHM, T. GNEITING, A. JORDAN et F. KRÜGER. "Of quantiles and expectiles : consistent scoring functions, Choquet representations and forecast rankings". In : *Journal of the Royal Statistical Society Series B : Statistical Methodology* 78.3 (2016).

[11] J. GASTINGER, T. SZTYLER, L. SHARMA et A. SCHUELKE. "On the Evaluation of Methods for Temporal Knowledge Graph Forecasting". In : *NeurIPS Temporal Graph Learning Workshop*. 2022.

[12] G. A. GESESE, R. BISWAS, M. ALAM et H. SACK. "A survey on knowledge graph embeddings with literals : Which model links better literal-ly ?" In : *Semantic Web* 12.4 (2021), p. 617-647.

[13] W. L. HAMILTON, R. YING et J. LESKOVEC. "Representation learning on graphs : Methods and applications". In : *arXiv preprint :1709.05584* (2017).

[14] K. HE, X. ZHANG, S. REN et J. SUN. "Delving deep into rectifiers : Surpassing human-level performance on imagenet classification". In : *Proceedings of the IEEE international conference on computer vision*. 2015, p. 1026-1034.

[15] P. HITZLER, M. KRÖTZSCH et S. RUDOLPH. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

[16] A. HOGAN et al. "Knowledge Graphs". In : *ACM Computing Surveys* 54.4 (juill. 2021).

[17] R. KALIYAPERUMAL et al. "Semantic modelling of common data elements for rare disease registries, and a prototype workflow for their deployment over registry data". In : *Journal of Biomedical Semantics* (mars 2022).

[18] A. KEEDY. "An overview of intracranial aneurysms". In : *McGill Journal of Medicine : MJM* 9.2 (2006), p. 141.

[19] D. P. KINGMA. "Adam : A method for stochastic optimization". In : *arXiv preprint :1412.6980* (2014).

[20] T. N. KIPF et M. WELLING. "Semi-Supervised Classification with Graph Convolutional Networks". In : *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net, 2017.

[21] A. KRISTIADI et al. "Incorporating literals into knowledge graph embeddings". In : *The Semantic Web–ISWC 2019 : 18th International Semantic Web Conference, Proceedings*. 2019, p. 347-363.

[22] M. LEHNE, S. LUIJTEN, P. VOM FELDE GENANNT IMBUSCH et S. THUN. "The use of FHIR in digital health–a review of the scientific literature". In : *German Medical Data Sciences : Shaping Change–Creative Solutions for Innovative Medicine* (2019).

[23] E. PRUD'HOMMEAUX et al. "Development of a FHIR RDF data transformation and validation framework and its evaluation". In : *Journal of Biomedical Informatics* 117 (2021), p. 103755.

[24] P. RISTOSKI et H. PAULHEIM. "Rdf2vec : Rdf graph embeddings for data mining". In : *International semantic web conference*. 2016, p. 498-514.

[25] M. SCHLICHTKRULL et al. "Modeling relational data with graph convolutional networks". In : *The semantic web : 15th international conference, ESWC 2018*. 2018, p. 593-607.

[26] V. TOURÉ et al. "FAIRification of health-related data using semantic web technologies in the Swiss Personalized Health Network". In : *Scientific Data* 10 (2023).

[27] E. A. VOSS et al. "Feasibility and utility of applications of the common data model to multiple, disparate observational health databases". In : *Journal of the American Medical Informatics Association* 22.3 (2015), p. 553-564.

[28] Q. WANG, Z. MAO, B. WANG et L. GUO. "Knowledge graph embedding : A survey of approaches and applications". In : *IEEE transactions on knowledge and data engineering* 29.12 (2017), p. 2724-2743.

[29] G. XIAO et al. "FHIR-Ontop-OMOP : Building clinical knowledge graphs in FHIR RDF with the OMOP Common data Model". In : *Journal of Biomedical Informatics* 134 (2022), p. 104201.

[30] F. ZHANG, Z. LI, D. PENG et J. CHENG. "RDF for temporal data management–a survey". In : *Earth science info.* 14 (2021), p. 563-599.

Session 2 : IA hybride et intégration neuro-symbolique

Exploitation de modèles neuronaux siamois pour l'amélioration du cadre Description-Détection de phénomènes épidémiologiques

Gabriel Henrique Alencar Medeiros[1*], Safaa Menad[1*], Lina F. Soualmia[1]

[1] Univ Rouen Normandie, INSA Rouen Normandie, Normandie Univ
LITIS UR 4108, FR-76000 Rouen, France

Résumé

Les systèmes de surveillance basés sur des événements (EBS) détectent les menaces sanitaires, mais souffrent d'une dépendance à l'expertise humaine. Le DDF (Description-Detection Framework) exploite PropaPhen, UMLS et OpenStreetMaps pour identifier des cas suspects à partir de données spatiotemporelles et textuelles, mais ne permet pas de classer ces observations. Pour pallier cette limite, nous intégrons BioSTransformers afin d'effectuer un clustering sémantique des observations. Cette approche améliore la classification en capturant les relations biomédicales et contextuelles, surpassant les méthodes classiques. Nos résultats montrent une réduction de la charge manuelle et une meilleure précision.

Mots-clés

Surveillance de la santé publique, Modèles de langues biomédicaux, Raisonnement spatiotemporel, cadre Description-Détection, Ontologie des phénomènes de propagation

Abstract

Event-based surveillance (EBS) systems detect health threats but rely heavily on human expertise. The DDF leverages PropaPhen, UMLS, and OpenStreetMaps to identify suspicious cases based on spatiotemporal and textual data but lacks the ability to classify these observations. To address this limitation, we integrate BioSTransformers to perform semantic clustering of observations. This approach enhances classification by capturing biomedical and contextual relationships, outperforming traditional methods. Our results demonstrate a reduction in manual effort and improved accuracy.

Keywords

Public Health Surveillance, Biomedical Language Models, Spatiotemporal Reasoning, Description-Detection Framework, Core Propagation Phenomenon Ontology

1 Introduction

Dans un monde de plus en plus vulnérable aux épidémies de grande ampleur et à propagation rapide, telles que le syndrome respiratoire aigu sévère, la maladie à virus Ebola et le coronavirus, des systèmes d'alerte précoce sensibles capables de détecter rapidement les menaces potentielles pour la santé publique avant qu'elles ne deviennent grandes et incontrôlables sont impératifs.

La surveillance basée sur les événements (EBS) peut être un élément clé d'un système d'alerte précoce efficace, permettant aux pays de mieux se préparer aux épidémies endémiques et pandémiques. Les systèmes EBS sont une approche organisée qui se concentre sur la détection rapide des événements de santé potentiels à partir de diverses sources de données. Ces sources comprennent les rapports de presse en ligne, les médias sociaux et d'autres plateformes numériques où les événements liés à la santé peuvent être discutés. L'objectif principal de l'EBS est d'identifier les premiers signaux de menaces sanitaires émergentes, permettant ainsi de réagir rapidement pour atténuer les éventuelles épidémies [2]. Ce processus commence par la détection de signaux ou d'observations qui alertent la communauté de santé publique sur la possibilité d'un événement dans une population.

Malgré ses avantages pour la détection précoce des épidémies, les systèmes EBS font face à plusieurs défis. Une limitation majeure est le besoin d'une intervention d'experts pour déterminer quels événements doivent être capturés et quels termes pertinents doivent être utilisés dans le processus de recherche [3]. Cette supervision manuelle peut entraîner des retards et des biais, limitant ainsi la capacité du système à s'adapter automatiquement aux menaces émergentes.

Une autre limitation est que de nombreuses applications EBS ne prennent pas souvent en compte les caractéristiques spatiales et temporelles, qui sont cruciales pour suivre la propagation des maladies et identifier les modèles d'épidémies au fil du temps [7]. L'absence d'analyse spatiotemporelle [†] peut entraîner une surveillance incomplète et réduire la prise de décision en matière de santé publique.

De nombreux systèmes EBS montrent une incapacité à traiter et à intégrer efficacement plusieurs sources de données hétérogènes non officielles au même temps, ce qui constitue une limitation importante. Avec l'augmentation du volume d'informations et des nouveaux types de médias disponibles via Internet, le domaine a évolué pour inclure des

∗. Ces auteurs ont contribué de manière égale à ce travail

†. Comportement d'un phénomène qui évolue dans le temps et l'espace, comme la propagation d'un virus.

types de données variés, tels que des rapports de presse, des publications sur les médias sociaux, des dossiers de santé officiels et des rapports publics informels. Ces sources varient en termes de structure, de format et de fiabilité, ce qui rend difficile l'extraction cohérente d'informations significatives [7].

Pour surmonter ces limitations, des recherches récentes ont proposé de nouvelles approches intégrant des techniques automatisées d'apprentissage machine et d'analyse géospatiale pour améliorer la précision des systèmes EBS. Parmi ces systèmes, le module DDF [11] traite la plupart des problèmes des EBS en détectant, décrivant et surveillant les cas suspects. Cependant, il n'est toujours pas capable de prédire de nouvelles menaces. Les méthodes de clustering peuvent être appliquées pour prédire des menaces inconnues. Cependant, les cas suspects doivent d'abord être représentés sous forme de vecteurs de valeurs réelles afin d'être traités efficacement. Il est nécessaire de transformer ces cas en représentations numériques.

Plusieurs méthodes peuvent être appliquées pour aider au clustering des cas suspects, comme l'approche de Wang [17], les graphes d'enchevêtrement et d'autres. Cependant, ces méthodes ne prennent pas en compte le contexte sémantique des termes biomédicaux, ce qui est crucial pour identifier précisément les menaces sanitaires.

Dans cet article, nous proposons d'intégrer BioSTransformers [15], un modèle de langue biomédical basé sur un transformeur de phrases, entraîné sur des données biomédicales de PubMed [‡]. Le modèle peut prédire la similarité sémantique entre les observations de différents cas, ce qui le rend particulièrement utile pour détecter les menaces sanitaires émergentes. Cet article est structuré comme suit : La Section 2 présente les travaux relatifs aux systèmes EBS. La Section 3 détaille notre approche pour améliorer le DDF. La Section 4 présente les résultats obtenus avec BioSTransformers et d'autres méthodes issues de la littérature et compare les résultats. Enfin, la Section 5 résume notre travail et expose les perspectives futures.

2 Travaux

Pour répondre aux limitations mentionnées des EBS, des approches automatisées intégrant des méthodologies basées sur la connaissance et basées sur les données ont gagné en importance.

2.1 Modèles EBS basés sur la connaissance vs. basés sur les données

Les modèles EBS basés sur la connaissance exploitent des ontologies structurées et des graphes de connaissances pour améliorer l'interprétabilité et le raisonnement. PropaPhen, un système basé sur une ontologie, intègre les dynamiques spatiotemporelles de propagation avec des connaissances de domaine, offrant ainsi un cadre explicable pour la surveillance des événements [10]. De même, le Unified Medical Language System (UMLS) et OpenStreetMaps (OSM) ont été utilisés pour améliorer la contextualisation géographique et biomédicale dans les données de surveillance [4, 5]. En revanche, les approches basées sur les données, telles que la classification de texte et le clustering basés sur l'apprentissage machine, reposent sur la découverte de motifs, mais manquent souvent de profondeur sémantique [19].

2.2 Le Description-Detection Framework

Le Description-Detection Framework (DDF) a été développé pour automatiser l'EBS en extrayant des observations structurées à partir de sources textuelles hétérogènes à l'aide de techniques d'extraction de relations. DDF construit un graphe de connaissances à partir de UMLS et OpenStreetMaps pour détecter des cas de santé suspects. Toutefois, il n'intègre pas de mécanisme pour classer ou prédire les motifs d'épidémies. Pour pallier cette limite, nous proposons d'y intégrer un processus de **clustering non supervisé**, permettant de regrouper automatiquement les cas suspects selon leur similarité sémantique. L'objectif est d'identifier des groupes cohérents d'observations potentiellement liées à un même phénomène épidémique, sans catégories prédéfinies, ce qui est essentiel dans un contexte où les virus émergents ne disposent pas encore de labels ou classifications établies.

2.3 Méthodes de clustering dans la surveillance de la santé publique

De nombreuses méthodes de clustering peuvent être proposées pour catégoriser les observations liées à la santé. La méthode de Wang, une approche de clustering hiérarchique largement utilisée, a été appliquée aux jeux de données épidémiologiques pour regrouper les cas similaires en fonction des symptômes extraits et des co-occurrences d'événements [17]. Cependant, les plongements ou représentations vectorielles (*embeddings*) traditionnels (par exemple, les plongements de graphes homogènes, le modèle Bag-of-Words) n'arrivent souvent pas à capturer les sémantiques biomédicales essentielles pour comprendre les relations entre les maladies [18].

2.4 Clustering basé sur les transformeurs pour l'EBS biomédical

Au cours des dernières années, plusieurs modèles de langue biomédicaux ont été développés, y compris BioBERT, PubMedBERT [6], etc. Bien que ces modèles aient démontré de bonnes performances dans diverses tâches de NLP biomédical, ils reposent principalement sur des architectures de cross-encoder, qui peuvent devenir coûteuses en termes de calculs et inefficaces lors du traitement de grands volumes de textes[16]. En revanche, les BioSTransformers [15] exploitent une architecture de bi-encodeur, permettant de traiter efficacement de grandes quantités de données textuelles biomédicales tout en maintenant des performances élevées. Ces modèles ont prouvé leur efficacité dans plusieurs tâches de similarité bio-sémantique, notamment l'alignement d'ontologies [14], la classification de documents biomédicaux, la recherche d'informations [12] et même l'enrichissement d'ontologies [13].

‡. https ://pubmed.ncbi.nlm.nih.gov/

3 Approche

3.1 Application biomédicale améliorée de DDF

Alors que les approches traditionnelles nécessitent un effort manuel considérable, les cadres basés sur les ontologies, tels que **PropaPhen**, ont permis un raisonnement automatisé sur les phénomènes de propagation spatiotemporels [10]. Le DDF exploite **PropaPhen, UMLS** et **OpenStreetMaps** pour extraire et structurer des observations liées à la santé à partir de sources de données textuelles [11]. Cependant, bien que DDF détecte efficacement les menaces potentielles, il ne possède pas de mécanismes pour classifier ces observations en catégories d'événements de santé significatives. Pour remédier à cette limitation, nous proposons une version améliorée, DDF+, qui intègre des techniques de clustering pour prendre en charge la classification automatique. PropaPhen est une ontologie conçue pour modéliser la propagation de phénomènes à travers des **domaines spatiotemporels et en réseau**. Elle repose sur des standards bien établis tels que **UFO-AB, OWL-Time, GeoSPARQL** et **SEAS** afin de fournir une représentation structurée des événements, des occurrences et de leurs interrelations. Cependant, sous sa forme générique, PropaPhen manque de spécificité pour le domaine biomédical.

Afin d'améliorer son applicabilité à la **surveillance de la santé publique**, nous introduisons **BioPropaPhen**, une extension spécialisée de PropaPhen à travers le module de description de DDP. BioPropaPhen intègre des connaissances du *Unified Medical Language System (UMLS)* et du *World Knowledge Graph (WorldKG)*, lui permettant ainsi de mieux représenter les concepts médicaux, les occurrences de maladies et leurs relations spatiotemporelles [9]. Sur la base de BioPropaPhen, nous construisons **BioPropaPhenKG**, un **graphe de connaissances** dynamique qui intègre :

- **Connaissances biomédicales** issues de UMLS et WorldKG pour structurer les entités et relations liées aux maladies.
- **Informations géospatiales** provenant d'OpenStreetMaps afin de contextualiser les observations dans les zones affectées.

Ce graphe de connaissances [§] constitue la base du cadre amélioré, facilitant l'extraction et la classification des observations liées aux maladies.

Le **module de détection** de DDF+ exploite des données textuelles hétérogènes via des **techniques d'extraction de relations**. Il traite les différentes sources : **articles de presse, rapports médicaux et les textes issus des réseaux sociaux** officielles et non officielles pour extraire les mentions d'événements sanitaires suspects et leurs liens spatiotemporels au niveau des phrases, des paragraphes ou des documents. Les observations extraites sont ensuite associées à BioPropaPhenKG, **enrichissant ainsi le graphe de connaissances** en intégrant les foyers de maladies détectés et leurs occurrences géotemporelles.

Pour passer de la détection à la classification automatique, DDF+ intègre des méthodologies de clustering. Chaque observation est transformée en une représentation numérique pour faciliter le clustering. Nous employons deux **méthodes de clustering de référence** à titre de comparaison :

- **Méthode de clustering de Wang :** Une approche établie dans la détection des foyers épidémiques, qui regroupe les observations selon des modèles de co-occurrence [17, 8]. Elle applique une formule de similarité biomédicale à chaque paire d'observations pour construire une **matrice de similarité**, utilisée ensuite pour le clustering.
- **Représentations homogènes par plongement :** Plongements générés en deux étapes : (1) extraction d'arêtes reliant des entités d'un graphe de connaissances existant à partir de textes, puis (2) application d'un modèle de plongement homogène tel que *Node2Vec*, qui projette le graphe dans un espace vectoriel unique, sans distinction de types de nœuds ou de relations. Le clustering est ensuite effectué par *K-means*, en utilisant la **distance euclidienne**.

Ces approches de base servent de point de référence pour évaluer l'impact de la sémantique biomédicale sur le clustering. Le cadre amélioré DDF+ est illustré à la Figure 1.

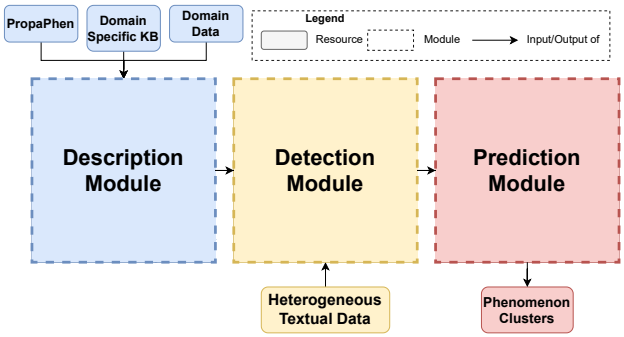

FIGURE 1 – Vue d'ensemble du cadre Description-Détection amélioré (DDF+).

Pour évaluer l'efficacité du clustering, nous validons les résultats à l'aide du **F1-score**. La vérité terrain est construite de la manière suivante :

- Chaque observation provient d'un ensemble de données textuelles spécifique associé à une **catégorie de maladie** connue (par exemple, COVID-19, variole du singe).
- Ces étiquettes de données servent de **clusters de référence** contre lesquels les clusterings prédits sont comparés.

L'évaluation mesure la précision du clustering des observations en fonction de leur source textuelle d'origine.

Bien que les méthodes de clustering de référence permettent une classification initiale, elles ne tirent pas parti de la sémantique biomédicale. À l'instar de la méthode de Wang, **BioSTransformers** calcule la similarité entre chaque paire d'observations pour construire une **matrice**

§. L'ensemble des données de BioPropaPhenKG est accessible à https://zenodo.org/records/10911980.

de distance, mais en s'appuyant sur des plongements sémantiques enrichis, issus de textes biomédicaux. Ce modèle de type transformeur, entraîné sur des corpus spécialisés, permet ainsi un clustering plus pertinent en intégrant des connaissances spécifiques au domaine. La section suivante détaille cette approche et illustre comment elle améliore la précision de la classification.

3.2 BioSTransformers

BioSTransformers [14] est un modèle de langue biomédical basé sur l'architecture Sentence-Transformer (transformeur de phrases), spécialement conçu pour capturer les relations sémantiques entre les termes biomédicaux. Il est entraîné sur les corpus biomédicaux de PubMed afin d'améliorer sa compréhension du langage spécifique au domaine. Initialement conçus pour associer des phrases de taille similaire à des représentations vectorielles, les transformeurs siamois bi-encodeur dans notre approche sont adaptés pour projeter les termes MeSH[¶], les titres et les résumés des articles PubMed dans un espace vectoriel commun. L'objectif est d'assurer une correspondance contextuelle dans cet espace.

3.2.1 Entraînement

Nos modèles ont été entraînés sur des articles de PubMed, à partir desquels nous avons extrait des paires de titres/résumés et leurs termes MeSH associés. Pour optimiser l'entraînement, nous avons employé la fonction de perte contrastive Multiple Negative Ranking Loss (MNRL), spécialement conçue pour entraîner des transformeurs siamois bi-encodeurs. Pour construire BioSTransformers, nous nous sommes inspirés de Sentence-BERT (SBERT) [16], en remplaçant BERT par d'autres transformeurs. Plus précisément, nous avons utilisé des transformeurs biomédicaux pré-entraînés sur des corpus biomédicaux et les avons adaptés en transformeurs siamois en intégrant une couche de regroupement (*pooling*) et en modifiant la fonction objectif. nous avons sélectionné les meilleurs transformeurs biomédicaux en termes de performance : BlueBERT, PubMedBERT, BioELECTRA et BioClinicalBERT [6, 1]. Nous avons ensuite développé des sentence-transformers sur ces transformeurs, donnant naissance aux modèles suivants : S-BlueBERT, S-PubMedBERT, S-BioELECTRA et SBio_ClinicalBERT. La couche de regroupement calcule le vecteur moyen des embeddings de sortie du transformeur. Les deux textes en entrée sont successivement traités par le transformeur, produisant deux vecteurs, u et v, qui sont ensuite transmis à la fonction objectif.

3.2.2 Modèles

Dans ce travail, nous avons selectionné le modèle SBio_ClinicalBERT car il a obtenu les meilleures performances en termes de score F1 par rapport aux autres modèles. Ce modèle a été appliqué pour comparer les observations et générer une matrice de scores de similarité, qui a ensuite été exploitée pour construire des clusters en fonction des similarités obtenues.

¶. https ://www.nlm.nih.gov/mesh/meshhome.html

4 Résultats

4.1 Présentation du jeu de données

Pour évaluer l'efficacité de l'approche de clustering proposée, nous avons utilisé six jeux de données couvrant **deux grands phénomènes de santé publique : COVID-19 et la variole du singe**. Ces jeux de données, résumés dans le Tableau 1, incluent des sources textuelles provenant **d'actualités en ligne, d'articles médicaux et des réseaux sociaux**, garantissant ainsi une représentation diversifiée des données de surveillance en conditions réelles.

Les jeux de données ont été soigneusement sélectionnés pour refléter **les dynamiques spatio-temporelles** dans les rapports de santé publique. Chaque jeu de données correspond à une période spécifique, permettant ainsi au cadre d'analyse de traiter et détecter les observations sanitaires dans un contexte temporel bien défini. Cette diversité permet une évaluation approfondie des méthodes de clustering sous différentes structures linguistiques et styles de rapport.

4.2 Évaluation des performances de clustering

Nous avons évalué trois approches de clustering : la méthode de clustering de Wang, les plongements homogènes et BioSTransformers, sur différents niveaux d'extraction de relations : correspondance de documents, de paragraphes et de phrases. Le Tableau 2 présente les scores F1 obtenus par chaque méthode pour chaque niveau d'extraction. Les résultats mettent en évidence unavantage clair de BioSTransformers par rapport aux méthodes traditionnelles.

Bien que **la méthode de Wang** ait démontré une précision raisonnable dans la détection des épidémies, ses performances restent limitées en raison de l'absence de **compréhension sémantique**. De même, les **plongements homogènes** n'ont pas réussi à capturer **les relations biomédicales spécifiques au domaine**, entraînant un clustering moins précis. En revanche, **BioSTransformers a exploité efficacement la sémantique biomédicale**, obtenant les scores F1 les plus élevés à tous les niveaux d'extraction. De plus, les **informations spatiales et temporelles ont été concaténées aux vecteurs d'observation**, en ajoutant les plongements des localisations et les valeurs temporelles, ce qui a permis d'enrichir la représentation contextuelle des cas et d'améliorer la qualité des clusters formés.

Ces résultats montrent que BioSTransformers améliore significativement la précision de classification en intégrant le contexte biomédical dans le clustering, le rendant ainsi plus adapté à la détection d'événements de santé publique.

‖. https://aylien.com/resources/datasets/coronavirus-dataset
**. https://allenai.org/data/cord-19
††. https://paperswithcode.com/dataset/the-reddit-covid-dataset
‡‡. https://doi.org/10.3390/idr14060087

Phénomène	Jeu de données	Période	Documents	Source
COVID-19	Aylien ‖	Nov-2019	8	Actualités en ligne
COVID-19	CORD-19 **	Déc-2019	726	Articles médicaux
COVID-19	Reddit COVID Dataset ††	Fév-2020	4,908	Réseaux sociaux
Variole du singe	Extrait de BBC News	Mai-2022	27	Actualités en ligne
Variole du singe	Extrait de PubMed	Juin-2022	36	Articles médicaux
Variole du singe	MonkeyPox2022 ‡‡	Mai-2022	33,826	Réseaux sociaux

TABLE 1 – Jeux de données sélectionnés pour l'expérimentation.

Méthode	documents	paragraphes	phrases
Clutering de Wang	0.68	0.75	0.62
Node2Vec	0.75	0.78	0.55
BioSTransformers	**0.9**	**0.79**	**0.71**

TABLE 2 – Scores F1 pour chaque méthode de clustering aux différents niveaux d'extraction des relations par niveau.

4.3 Visualisation du meilleur résultat de clustering

Pour illustrer l'efficacité de BioSTransformers, la Figure 2 présente un **nuage de points du meilleur clustering obtenu pour l'extraction de relations au niveau document**, visualisé à l'aide de **t-SNE**. Les couleurs indiquent la **vérité terrain** (COVID-19 en bleu, variole du singe en orange), tandis que les lettres C et M représentent les **prédictions du clustering**. On observe une **séparation nette des phénomènes** grâce aux **plongements de BioSTransformers**. Contrairement aux méthodes de référence qui ont produit des **clusters qui se chevauchent** en raison d'une faible prise en compte de la sémantique, BioSTransformers a permis une **séparation distincte des observations liées aux maladies**. Cette visualisation confirme les **capacités supérieures de classification** de notre approche.

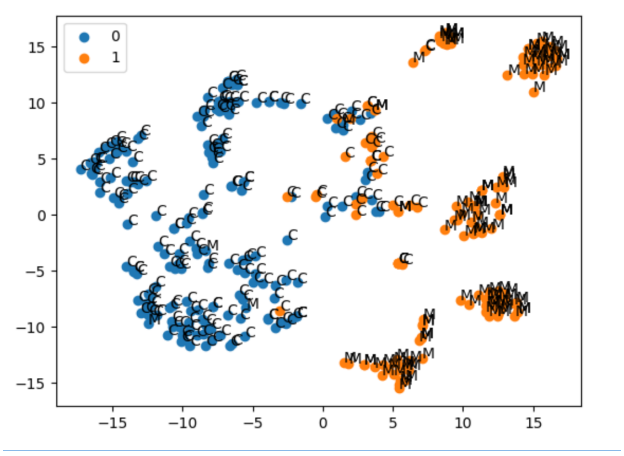

FIGURE 2 – Visualisation du clustering du meilleur résultat obtenu avec BioSTransformers. La séparation distincte des observations liées aux maladies confirme son efficacité. Les clusters sont représentés par des couleurs, et les initiales des maladies représentent la vérité terrain.

4.4 Discussion

Les résultats confirment que l'intégration de connaissances sémantiques biomédicales dans les plongements améliore significativement le clustering des observations de santé. La comparaison entre la méthode de Wang, les plongements homogènes et BioSTransformers souligne l'importance d'une contextualisation fine et spécifique au domaine. Toutefois, certaines limites subsistent : le système se limite à des documents en anglais, ce qui réduit l'accès à des signaux précoces locaux dans d'autres langues. Il a été testé uniquement sur deux phénomènes sanitaires (COVID-19 et variole du singe), et son extension à d'autres épidémies, vagues ou variants reste à explorer. En outre, l'absence de module de détection des fausses informations rend le clustering sensible à la désinformation. Ces limites ouvrent des perspectives d'amélioration. Les scores F1 obtenus avec BioSTransformers démontrent l'efficacité de l'approche dans un contexte de surveillance épidémiologique. Cette étude met en évidence le potentiel des modèles NLP avancés pour automatiser et enrichir le clustering biomédical. Le code source est disponible sur le dépôt Git du projet .

5 Conclusion

Dans ce travail, nous avons proposé une approche robuste intégrant la correspondance sémantique biomédicale au sein du cadre DDF, grâce à l'utilisation de BioSTransformers. Basé sur PropaPhen, UMLS et OpenStreetMaps, ce système détecte des cas sanitaires suspects à partir de données spatiotemporelles et textuelles. Notre contribution enrichit ce cadre en y ajoutant la classification automatique des observations. L'intégration de BioSTransformers a permis un clustering plus précis en combinant sémantique biomédicale et contexte spatiotemporel, surpassant les méthodes traditionnelles de la littérature. Grâce à des transformeurs de phrases, nous avons obtenu des résultats compétitifs dans la capture des relations bio-sémantiques à travers différents types de textes. Pour les travaux futurs, nous prévoyons de tester ce cadre sur d'autres épidémies ainsi que sur différentes vagues ou variants d'un même virus, afin d'analyser des dynamiques plus fines. Nous comptons également étendre l'analyse à des documents multilingues, en incluant des langues locales pour renforcer la détection de signaux précoces propres à chaque pays. D'autres pistes incluent l'exploration de méthodes de NLP hors LLM, et l'in-

https ://github.com/Gabriel382/DDPF-Health-Risks

tégration de représentations sémantiques hétérogènes ou de connaissances ontologiques dans les plongements.
Enfin, pour accroître la robustesse du système, nous envisageons l'ajout d'un module de détection des fausses informations, afin de limiter l'impact des contenus erronés sur le clustering. Ces améliorations visent un système de surveillance plus fiable, explicable et adaptable, confirmant le potentiel des transformeurs pour la santé publique et la surveillance épidémiologique.

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] S Arunmozhi Balajee, Stephanie J Salyer, Blanche Greene-Cramer, Mahmoud Sadek, and Anthony W Mounts. The practice of event-based surveillance : concept and methods. *Global Security : Health, Science and Policy*, 6(1) :1–9, 2021.

[3] Philippe Barboza, Laetitia Vaillant, Abla Mawudeku, Noele P Nelson, David M Hartley, Lawrence C Madoff, Jens P Linge, Nigel Collier, John S Brownstein, Roman Yangarber, et al. Evaluation of epidemic intelligence systems integrated in the early alerting and reporting project for the detection of a/h5n1 influenza events. *PLoS One*, 8(3) :e57252, 2013.

[4] O Bodenreider. The unified medical language system (UMLS) : integrating biomedical terminology. *Nucleic Acids Res.*, 32(90001) :267D–270, January 2004.

[5] Alishiba Dsouza, Nicolas Tempelmeier, Ran Yu, Simon Gottschalk, and Elena Demidova. WorldKG : A world-scale geographic knowledge graph. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, New York, NY, USA, October 2021. ACM.

[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] David M Hartley, Noele P Nelson, RR Arthur, Philippe Barboza, Nigel Collier, Nigel Lightfoot, JP Linge, E van der Goot, Abla Mawudeku, LC Madoff, et al. An overview of internet biosurveillance. *Clinical Microbiology and Infection*, 19(11) :1006–1013, 2013.

[8] Lucas Maddalena and Fernanda Baião. An application of the disease ontology (do) for clustering covid-19 hospitalizations in rio de janeiro, 2024.

[9] Gabriel H A Medeiros, Lina F Soualmia, and Cecilia Zanni-Merk. Biopropaphenkg : A biomedical knowledge graph for event-based surveillance. *Journal of Medical AI Research*, 2024.

[10] Gabriel H A Medeiros, Lina F Soualmia, and Cecilia Zanni-Merk. Harnessing the core propagation phenomenon ontology to develop a knowledge graph for tracking health-related phenomena. *Stud. Health Technol. Inform.*, 316 :1933–1937, August 2024.

[11] Gabriel H A Medeiros, Lina F Soualmia, and Cecilia Zanni-Merk. Towards public health-risk detection and analysis through textual data mining. *Procedia Comput. Sci.*, 246 :3014–3023, 2024.

[12] Safaa Menad, Saïd Abdeddaïm, and Lina F Soualmia. Flexible classification, question-answering and retrieval with siamese neural networks for biomedical texts. In *International Conference on Flexible Query Answering Systems*, pages 27–38. Springer, 2023.

[13] Safaa Menad, Saïd Abdeddaïm, and Lina F Soualmia. Simhomer : Siamese models for health ontologies merging and validation through large language models. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 117–129. Springer, 2024.

[14] Safaa Menad, Wissame Laddada, Saïd Abdeddaïm, and Lina F Soualmia. Biostransformers for biomedical ontologies alignment. In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2023, Volume 2 : KEOD*, pages 73–84. SCITEPRESS, 2023.

[15] Safaa Menad, Wissame Laddada, Saïd Abdeddaïm, and Lina F Soualmia. New siamese neural networks for text classification and ontologies alignment. In *International Conference on Complex Computational Ecosystems*, pages 16–29. Springer, 2023.

[16] 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.

[17] James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. A new method to measure the semantic similarity of GO terms. *Bioinformatics*, 23(10) :1274–1281, May 2007.

[18] Yun Xiong, Mengjie Guo, Lu Ruan, Xiangnan Kong, Chunlei Tang, Yangyong Zhu, and Wei Wang. Heterogeneous network embedding enabling accurate disease association predictions. *BMC Med. Genomics*, 12(Suppl 10) :186, December 2019.

[19] Daniel Zeng, Zhidong Cao, and Daniel B Neill. Artificial intelligence–enabled public health surveillance—from local detection to global epidemic monitoring and control. In *Artificial Intelligence in Medicine*, pages 437–453. Elsevier, 2021.

Extraction terminologique juridique à faible supervision : une méthode hybride combinant LLM, règles syntaxiques et CamemBERT

Julien Breton[1,2], Mokhtar Boumedyen Billami[2], Max Chevalier[1], Cassia Trojahn[1]

[1] Institut de Recherche en Informatique de Toulouse, IRIT
[2] Berger-Levrault

julien.breton@irit.fr, mb.billami@berger-levrault.com,
max.chevalier@irit.fr, cassia.trojahn@irit.fr

Résumé

Le secteur juridique se caractérise par un nombre important de documents et par leur complexité. Les entreprises ont l'obligation d'appliquer ces dispositions juridiques. En raison de l'évolution constante de ces documents, un intérêt croissant se manifeste pour l'automatisation du traitement des textes juridiques afin de faciliter la conformité réglementaire. Une étape clé de ce processus réside dans l'extraction des termes juridiques. Les méthodes état de l'art, telles que les systèmes à base de règles, les réseaux Bi-LSTM et BERT, requièrent une quantité importante de données annotées pour atteindre des performances satisfaisantes, une tâche particulièrement chronophage pour les experts du domaine. Avec l'essor des grands modèles de langage (LLM), la recherche s'oriente de plus en plus vers l'exploitation de leurs capacités, notamment à travers des approches faiblement supervisées. Dans cet article, nous présentons un système hybride qui distille les connaissances de GPT-4 vers un modèle CamemBERT, tout en appliquant un filtrage syntaxique. Cette approche réduit non seulement le besoin d'intervention d'experts par rapport au système CamemBERT classique, mais elle surpasse également le système reposant uniquement sur GPT-4, en améliorant le score F1 de 7 à 24 points de pourcentage.

Mots-clés

Extraction terminologique juridique, Faible supervision, CamemBERT, Grands modèles de langage (LLM), Distillation des connaissances

1 Introduction

Le domaine juridique se caractérise par un volume considérable de documents en constante évolution, tels que les contrats, la législation, les décisions de justice ou encore les décrets. Ces documents sont denses, complexes et rédigés dans un langage hautement spécialisé, ce qui rend leur analyse et application à la fois chronophages et susceptibles aux erreurs humaines. Cependant, les entreprises ont l'obligation légale de se mettre en conformité avec ces dispositions juridiques, sous peine d'amendes. Comme le soulignent Sassier et al. [23], en France, « plus de 10 500 lois, 120 000 décrets, 7 400 traités, 17 000 textes communautaires, des dizaines de milliers de pages réparties dans 62 codes distincts » sont en vigueur. Certains de ces textes font d'ailleurs l'objet de modifications fréquentes : « 6 modifications par jour ouvrable pour le Code des impôts de 2006 ». C'est dans ce sens que la recherche vise à automatiser le traitement des documents juridiques. Elle souhaite non seulement accélérer leur analyse, tout en délestant les experts juridiques de cette tâche chronophage et à faible valeur ajoutée.

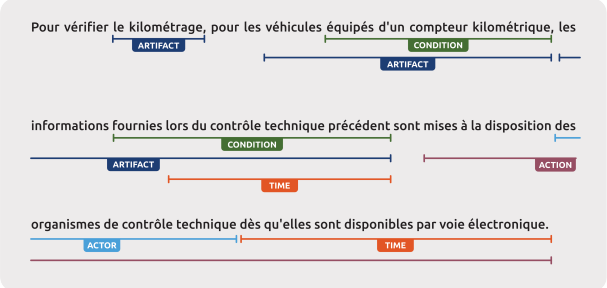

FIGURE 1 – Extraction des termes juridiques à partir de la phrase suivante : « Pour vérifier le kilométrage, pour les véhicules équipés d'un compteur kilométrique, les informations fournies lors du contrôle technique précédent sont mises à la disposition des organismes de contrôle technique dès qu'elles sont disponibles par voie électronique ».

La tâche fondamentale pour l'extraction des règles juridiques consiste à les formaliser de manière structurée. Deux tâches entrent alors en jeu : l'extraction des termes juridiques et l'extraction des relations entre ces termes. Le présent article se concentre sur l'extraction terminologique, comme l'illustre la Figure 1, qui provient d'un exemple issu du jeu de données utilisé dans cette étude.
Le jeu de données exploité dans le cadre de cette recherche a été introduit par Sleimi et al. [26], qui se sont appuyés sur des documents juridiques du Luxembourg. Dans leur étude, les auteurs ont développé un système fondé sur des

règles syntaxiques pour l'extraction de termes juridiques. Ils ont obtenu une précision notable de 0,874 et un rappel de 0,855. Cependant, atteindre de telles performances requiert un investissement conséquent en temps de la part d'experts pour annoter les données et concevoir les règles syntaxiques. D'autres méthodes, telles que les LSTM (Long Short-Term Memory) [24] ou BERT (Bidirectional Encoder Representations from Transformers) [17], rencontrent les mêmes contraintes liées aux données annotées. Néanmoins, l'émergence des grands modèles de langage (LLMs) [36] ouvre de nouvelles perspectives en réduisant l'implication humaine requise pour l'extraction. En tirant parti de leur connaissance fondationnelle, les LLMs sont par exemple capables de réaliser de l'extraction d'entités nommées dans des documents en biologie [29].

Cet article propose une approche hybride combinant LLM, méthodes fondées sur des règles syntaxiques et modèle de langage. La distillation des connaissances fournies par un LLM vers un modèle CamemBERT, avec un filtrage basé sur des règles, répond à plusieurs objectifs. Premièrement, cette approche limite l'implication des experts à la formulation des instructions pour le LLM et à la définition des règles de filtrage. Deuxièmement, elle diminue les besoins en ressources de calcul et améliore l'interprétabilité des résultats, en offrant une meilleure compréhension du processus d'extraction ainsi que du jeu de données utilisé pour l'entraînement.

Les travaux présentés dans cet article sont l'adaptation en version française de l'article [4] paru lors de la conférence EKAW 2024.

Le reste de l'article est organisé comme suit : la section 2 présente les principaux travaux connexes ; la Section 3 décrit les modèles de référence utilisés (CamemBERT et GPT-4) ; la Section 4 introduit le système hybride proposé ; la Section 5 détaille le jeu de données utilisé ainsi que les résultats obtenus selon les différentes stratégies ; enfin, la Section 6 conclut l'article et propose des perspectives de recherche futures.

2 Travaux connexes

Le traitement des connaissances constitue un domaine vaste ayant suscité de nombreuses contributions de la part de la communauté scientifique. Récemment, une tendance marquée s'est manifestée en faveur de l'utilisation des réseaux de neurones pour des tâches d'analyse [28] ou de classification [6, 22]. Par exemple, Biener et al. [2] ont développé un système d'anonymisation d'entités nommées (Named Entity Recognition, NER [18]) dans des documents financiers rédigés en allemand. Leur système identifie des entités telles que les noms et prénoms, les adresses postales et électroniques, ainsi que les localisations. L'étude a évalué diverses architectures, notamment les RNN, LSTM et Conditional Random Fields (CRF) [15]. Les résultats démontrent que les architectures combinant RNN et CRF obtiennent les meilleures performances, avec un rappel de plus de 97 % sans post-traitement et environ 99 % après post-traitement, tout en maintenant une précision supérieure à 90 %. Malgré l'efficacité de cette approche, elle présente certaines limites en raison de la forte implication des experts et du caractère chronophage des tâches. L'annotation manuelle d'un corpus de 407 documents financiers allemands publiés, représentant un total de 189 000 tokens, constitue un travail particulièrement lourd et coûteux en temps.

Le modèle BERT, proposé par Vaswani et al. [30], vise à atténuer ce problème grâce à un entraînement initial sur un large corpus de données, comprenant environ 3,3 millions de mots issus du Toronto BookCorpus et de Wikipédia en anglais. Cet entraînement préliminaire confère au modèle une base de connaissances génériques qui peut ensuite être affinée sur des jeux de données de taille plus réduite. La communauté juridique a adopté cette architecture pour la reconnaissance d'entités nommées, comme en témoignent les travaux menés sur des décisions judiciaires italiennes [21] ou des textes juridiques brésiliens [32]. Ces approches associent BERT à des architectures de type Bi-LSTM et CRF. Toutefois, malgré leurs performances, ces méthodes nécessitent toujours un volume conséquent de données annotées et un investissement important d'expertise humaine.

Avec l'essor récent des grands modèles de langage (LLMs), de nombreuses études ont exploré leur potentiel pour l'extraction d'informations, obtenant des résultats significatifs [9, 33, 7, 1]. Des avancées notables ont aussi été accomplies dans la distillation des performances de ces modèles vers des modèles de plus petite taille [12]. Yuxian et al. [11] ont démontré que la distillation constitue une méthode efficace pour transférer les connaissances d'un grand modèle génératif vers un modèle plus compact. Ce processus favorise l'obtention de réponses plus précises ainsi qu'une amélioration des performances. Cependant, des travaux constatent des limites dans l'utilisation des LLM pour l'extraction d'informations, que ce soit dans les capacités de calcul [13, 31] ou même dans les résultats obtenus [14]. [10] montre que les LLM ne parviennent pas à surpasser les modèles traditionnels (BERT) dans les tâches de NER, soulignant les limites des LLM dans l'extraction d'entités complexes spécifiques à un domaine. Xie et al. [34] examinent les performances des LLM dans les tâches de NER non supervisées, notant que bien que des améliorations puissent être réalisées, des défis subsistent pour atteindre un haut niveau de précision sans recourir à des stratégies personnalisées.

Dans le domaine juridique, comme le souligne l'étude de Solihin et al. [27], les travaux se sont majoritairement concentrés sur la NER, ce qui a donné lieu à une recherche relativement limitée et à peu de jeux de données spécifiquement consacrés à l'extraction de termes juridiques. Bien que ces deux tâches puissent sembler similaires a priori, elles s'en distinguent considérablement. Les entités nommées, telles que les personnes ou les organisations, sont généralement représentées par une unité courte et bien délimitée, comme *« Stephen Hawking »*. À l'inverse, les entités juridiques englobent des concepts plus larges, des entités nommées ou même des expressions complètes. Par exemple, le concept juridique d'« Acteur » peut corres-

pondre à une entité comme *« le conducteur »*, qui n'est pas une entité nommée. De même, la notion de « Condition » peut inclure un contenu tel que *« si l'ensemble couplé de véhicules se compose de deux véhicules automoteurs »*. Ces exemples illustrent que notre tâche dépasse largement le cadre NER traditionnel et peut s'apparenter à l'extraction de segments textuels. Il est donc légitime de remettre en question l'efficacité directe des approches conçues pour la NER [37, 35, 19, 20] dans l'extraction de termes juridiques.

Concernant l'extraction de termes juridiques, certaines études, telles que celle de Sleimi et al. [25], ont développé leurs propres jeux de données et mis en œuvre une approche fondée sur des règles syntaxiques. Des travaux plus récents, tels que ceux de Castano et al. [5], explorent une démarche similaire en se focalisant sur l'extraction conjointe de concepts et d'entités à partir de documents juridiques européens. Les éléments extraits sont ensuite intégrés dans un système de gestion des connaissances. D'autres contributions, comme celle de Dragoni et al. [8], montrent la pertinence et l'efficacité d'une approche hybride pour l'extraction de règles juridiques à partir de documents textuels. Ces résultats suggèrent que la combinaison de plusieurs techniques peut considérablement améliorer la précision et l'efficacité de l'extraction, tout en compensant leurs défauts mutuels.

Basé sur ces observations, notre étude a pour objectif de comparer notre système hybride avec les modèles de référence, à savoir CamemBERT et GPT-4, qui seront présentés dans la section suivante.

3 Systèmes de référence

Dans le but de comparer notre approche aux systèmes actuels de l'état de l'art, nous introduisons les modèles de langage pour la tâche d'extraction de termes juridiques. Les sections suivantes décrivent leur mise en œuvre, tout en mettant en lumière les avantages et les limites propres à chacune de ces architectures.

3.1 Extraction terminologique juridique à l'aide de CamemBERT

En raison de la langue française du jeu de données utilisé (décrit en Section 5.1), nous avons opté pour un modèle CamemBERT, au lieu du modèle BERT original. Le modèle Legal-CamemBERT-base [16], spécifiquement réentraîné sur plus de 22 000 articles juridiques du droit belge en français, se révèle plus adapté au traitement et à la compréhension des textes juridiques en langue française. Ce choix vise à mieux capter les subtilités et les spécificités linguistiques présentes dans notre corpus, améliorant ainsi la représentation des plongements lexicaux.

L'extraction de termes juridiques avec CamemBERT requiert un corpus annoté manuellement par des experts. Ce jeu de données doit être divisé en deux parties : l'une pour l'entraînement et la seconde pour l'évaluation, comme indiqué à la Figure 2. L'extraction terminologique au moyen de CamemBERT repose traditionnellement sur le schéma d'annotation Inside-Outside-Beginning (IOB), cou-

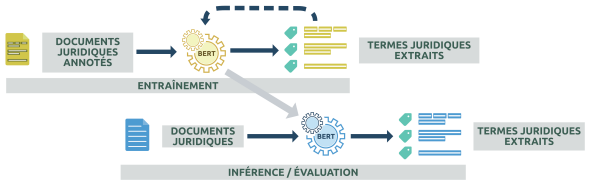

FIGURE 2 – Processus global basé sur CamemBERT : (i) réentraînement du modèle CamemBERT, (ii) évaluation de l'inférance du modèle réentraîné.

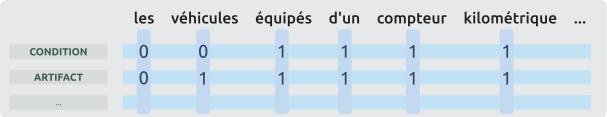

FIGURE 3 – Matrice de tokenisation basée sur la phrase courte : « les véhicules équipés d'un compteur kilométrique ». Les mots activent les concepts juridiques tels que `Condition` ou `Artifact` à l'aide de valeurs binaires (0 ou 1).

ramment utilisé en traitement automatique des langues pour l'étiquetage de séquences, notamment dans les tâches de NER. Toutefois, dans le cas de l'extraction de termes juridiques, des chevauchements d'annotations sont fréquemment observés, ce qui nécessite une adaptation de l'architecture interne du modèle. Par exemple, dans la Figure 1, nous constatons que le segment « équipés d'un compteur kilométrique » est à la fois catégorisé comme `Artifact` et comme `Condition`.

Une première modification porte donc sur la matrice d'entrée exploitée dans le modèle PyTorch, pour permettre une classification multi-label et multi-classe. La Figure 3 illustre cette modification, qui permet l'activation de plusieurs concepts juridiques pour un même token, comme pour le mot « compteur ». La seconde modification concerne la mesure utilisée lors de l'entraînement ; nous avons opté pour un macro score F1, c'est-à-dire une moyenne des scores F1 calculés pour chaque concept individuellement. Enfin, une dernière adaptation consiste à modifier le classifieur de sortie à l'aide de la bibliothèque Transformers de Huggingface. Cette modification substitue la fonction de perte CrossEntropyLoss [1] par la fonction BCEWithLogitsLoss [2], qui combine l'entropie croisée binaire à une fonction sigmoïde, convenant aux tâches multi-label. Cette architecture est schématisée par la Figure 4 et les expériences décrites dans cet article sont disponibles dans notre dépôt GitLab [3].

Il est important de mentionner que le réentraînement sur les 22 000 articles, réalisé pour créer le modèle Legal-CamemBERT-base [16], diffère des réentraînements réalisés dans notre étude. En effet, les articles du droit belge ont eu pour objectif d'améliorer la représentation sémantique,

1. https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html
2. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html
3. https://gitlab.irit.fr/ala/legal-concepts-extraction

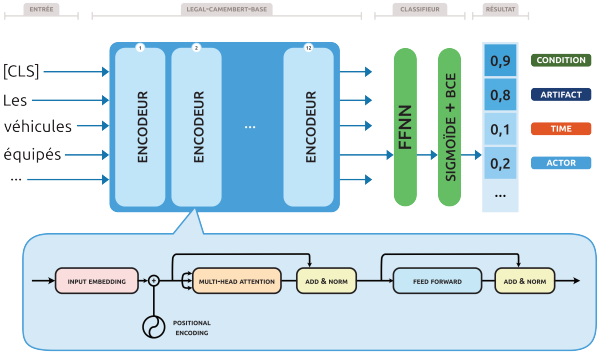

FIGURE 4 – Architecture du modèle CamemBERT utilisé dans le cadre de notre étude. Les modifications concernent le classifieur en sortie du modèle CamemBERT, incluant la fonction de perte.

tandis que nos réentraînements effectués via le corpus de Sleimi et al. [26] visent le classifieur de sortie, comme illustré dans la Figure 4.

L'entraînement du classifieur en sortie de CamemBERT nécessite un jeu de données conséquent dont l'élaboration représente une tâche fastidieuse pour les experts. L'apparition récente de modèles génératifs, tels que GPT-4, a ouvert de nouvelles voies de recherche autour de l'extraction de connaissances faiblement supervisée. Ces modèles sont en mesure d'effectuer des tâches d'extraction sans recourir à un grand volume de données annotées. Cette approche est détaillée dans la section suivante.

3.2 Extraction terminologique juridique fondée sur un LLM

La Figure 5 illustre le processus global d'utilisation d'un LLM pour l'extraction de termes à partir de textes juridiques. En employant l'ingénierie des requêtes (*prompt engineering*), nous utilisons le LLM pour une tâche d'extraction terminologique. L'un des principaux atouts de cette approche repose sur l'entraînement fondationnel du modèle, qui lui permet d'obtenir de bonnes performances à partir d'instructions minimales.

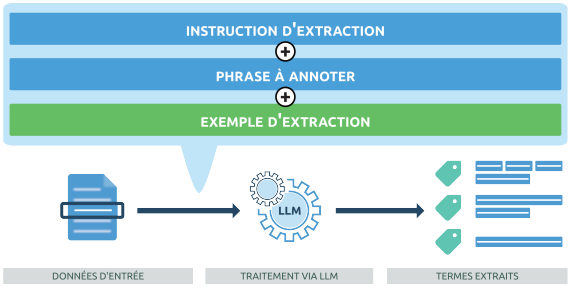

FIGURE 5 – Processus d'extraction terminologique employant un LLM via l'ingénierie des requêtes.

La structure de la requête utilisée a émergé d'un processus empirique et d'expérimentations itératives visant à identifier la formulation la plus robuste. Ce travail s'appuie sur les recommandations et bonnes pratiques proposées par OpenAI[4]. Des ajustements successifs ont été opérés sur les différentes composantes, incluant la définition du rôle du modèle, la description des tâches et le format de sortie attendu. Les requêtes, disponibles dans notre dépôt GitLab[5], sont organisées comme suit : en premier lieu, un rôle est assigné au modèle (par exemple « expert en TAL »), conjointement à la tâche visée (« extraire des termes à partir d'énoncés »). Ensuite, sont présentés les différents concepts juridiques accompagnés de leurs définitions. Ces définitions constituent la seule connaissance externe introduite, nécessitant une contribution minimale des experts. À cela s'ajoute un exemple d'extraction avec la sortie attendue (en JSON), fournissant ainsi un aperçu explicite de la tâche à effectuer. La Figure 5 illustre la structure des différentes composantes au sein de la requête fournie en entrée du LLM.

Après avoir présenté l'application des LLMs à l'extraction d'information par ingénierie de requête, nous décrivons à présent notre approche hybride combinant un LLM, un filtrage syntaxique et un CamemBERT.

4 Approche hybride

Comme introduit précédemment, les modèles de type BERT offrent une efficacité notable mais requièrent un jeu de données conséquent. À l'inverse, les LLMs, du fait de leur pré-entraînement, ne dépendent pas d'un corpus spécifique, mais peuvent souffrir de précision dans la délimitation des termes. Pour surmonter les limites de ces deux approches, nous avons développé une architecture hybride, illustrée par la Figure 6. Cette approche combine un LLM afin d'amorcer l'extraction, suivie de règles syntaxiques pour filtrer les hallucinations du LLM, et se termine par un modèle CamemBERT permettant d'apprendre et de compresser cette connaissance.

La première étape de notre chaîne de traitement consiste donc à amorcer une extraction à l'aide de GPT-4, identique au processus décrit en Section 3.2. Les experts élaborent une requête contenant les définitions des concepts juridiques, un exemple commenté, et l'énoncé cible à analyser. Le LLM génère ensuite la prédiction correspondante. Une fois ce corpus synthétique généré, les termes extraits par le LLM sont filtrés via des règles syntaxiques. L'objectif consiste à améliorer la précision des annotations, en s'appuyant sur 15 règles définies par des experts [25]. Par exemple, dans la phrase illustrée à la Figure 1, les concepts de type `Artifact` peuvent être associés à des groupes nominaux, tandis que `Action` relèvent de groupes verbaux. Cette étape permet de traiter la problématique des limites (*boundary issue*) et de garantir la qualité des termes annotés.

Après avoir filtré les termes juridiques, ce corpus est utilisé afin de réentraîner le classifieur CamemBERT, selon

4. https://platform.openai.com/docs/guides/prompt-engineering
5. https://gitlab.irit.fr/ala/legal-entity-extraction/-/raw/main/modules/llm/utils.py

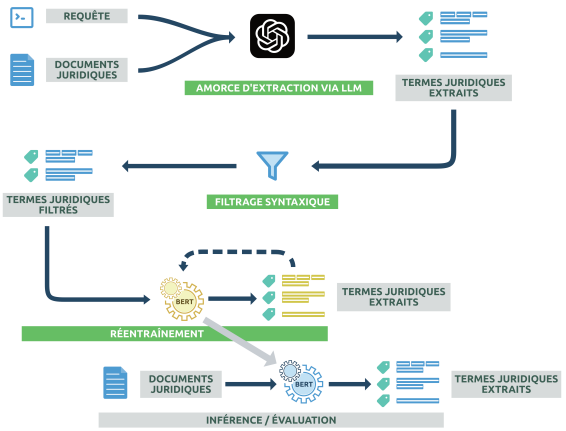

FIGURE 6 – Processus global de l'approche hybride : (i) amorçage de l'extraction terminologique via GPT-4, (ii) filtrage des résultats à l'aide de règles syntaxiques, (iii) réentraînement d'un CamemBERT.

les modalités présentées en Section 3.1. Le modèle est modifié par l'ajout d'une couche linéaire finale et l'ajustement de la fonction de perte adaptée au contexte multi-label. À l'issue de l'entraînement, le modèle est prêt à effectuer des extractions sur de nouveaux documents.

En distillant les connaissances d'un LLM dans un modèle CamemBERT, cette approche permet d'exploiter les atouts complémentaires des deux paradigmes, aboutissant à un système robuste et performant pour l'extraction de termes juridiques. La section suivante détaille les résultats obtenus et les compare avec les systèmes de référence.

5 Expérimentations et résultats

Après avoir présenté l'architecture hybride, nous nous intéressons à présent à l'évaluation de son efficacité et la comparons avec des systèmes de référence. Nous commençons par décrire le jeu de données utilisé dans l'ensemble de nos expérimentations, puis nous évaluons les performances du modèle CamemBERT seul, suivi du LLM avec GPT-4, et terminons avec les résultats du modèle hybride.

5.1 Jeu de données

Le jeu de données utilisé dans notre étude comprend 200 énoncés en français extraits du Code de la route luxembourgeois, annotés manuellement par des experts, identifiant 1339 segments au total [26]. L'annotation porte sur 14 concepts juridiques, et l'ensemble du corpus est disponible en ligne [6]. Dans le cadre de notre étude, nous nous concentrons sur un sous-ensemble de 8 concepts : `Action`, `Actor`, `Artifact`, `Condition`, `Location`, `Modality`, `Reference` et `Time`. Cette sélection correspond à des travaux antérieurs ayant conduit à la création du modèle sémantique SEMLEG [3], dédié à la formalisation de règles juridiques. Ce choix repose aussi sur le constat que certains concepts étaient sous-représentés dans les données d'origine. Afin de favoriser la réutilisabilité du corpus dans d'autres domaines applicatifs, nous avons choisi de nous limiter à ces huit concepts, définis dans le Tableau 1.

Concept	Définition
Action	Le processus de faire quelque chose.
Actor	Entité qui a la capacité d'agir.
Artifact	Objet matériel ou immatériel impliqué dans une action.
Condition	Une contrainte précisant les propriétés qui doivent être respectées.
Location	Un lieu où une action peut être réalisée.
Modality	Représente la contrainte d'application d'une règle.
Reference	Mention textuelle d'une autre source juridique.
Time	Moment, durée ou occurrence d'une action.

TABLE 1 – Définitions des huit concepts de [26] que nous utilisons dans notre étude.

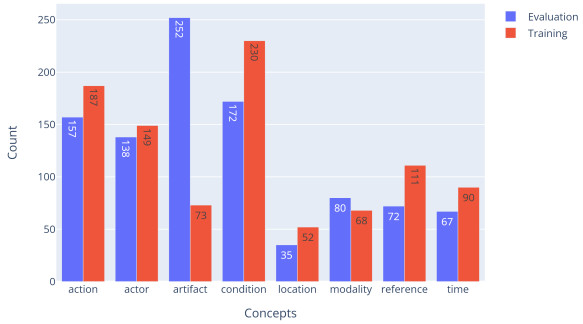

FIGURE 7 – Distribution des concepts juridiques dans les données d'entraînement et d'évaluation [26]. Huit concepts sont conservés : `Action`, `Actor`, `Artifact`, `Condition`, `Location`, `Modality`, `Reference` et `Time`.

La Figure 7 détaille la distribution des concepts retenus. Comme le montre cette distribution, le jeu de données issu de [26] présente un déséquilibre significatif entre les différentes classes. Un cas particulièrement notable concerne le concept `Artifact`, qui ne compte que 73 instances dans l'ensemble d'entraînement contre 252 dans l'ensemble d'évaluation. Ce déséquilibre peut influencer les performances des modèles d'apprentissage, en limitant leur capacité à généraliser sur des classes sous-représentées. Afin de garantir une comparaison avec l'approche des auteurs, nous avons choisi de conserver la distribution originale des annotations sans procéder à un rééquilibrage. Cette décision nous permet non seulement d'évaluer les performances de nos méthodes dans un contexte comparable, mais également d'analyser la robustesse de nos méthodes face à des situations dans lesquelles certaines catégories sont sous-représentées. Ainsi, cette configuration constitue une opportunité d'étudier dans quelle mesure l'approche hybride peut s'adapter aux contraintes inhérentes aux données déséquilibrées.

6. https://sites.google.com/view/metax-re2018/

5.2 Résultats du modèle CamemBERT réentraîné

Les performances du modèle d'extraction terminologique juridique fondé sur Legal-CamemBERT-base [16] sont présentées dans le Tableau 2. Celui-ci fournit les scores de précision, rappel et F1 pour chacun des huit concepts étudiés.

	Concepts juridiques							
	Action	Actor	Artifact	Condition	Location	Modality	Reference	Time
Précision	0.93	0.84	0.69	0.80	0.65	0.91	0.88	0.85
Rappel	0.41	0.60	0.16	0.83	0.53	0.48	0.67	0.66
F1	0.57	0.70	0.26	0.82	0.59	0.63	0.76	0.74

TABLE 2 – Résultats de précision, rappel et F1 obtenus avec Legal-CamemBERT-base [16] pour l'extraction de termes juridiques.

L'approche basée sur CamemBERT obtient un score F1 moyen supérieur à 0.63. La précision moyenne atteint 0.81, ce qui montre que le modèle extrait les termes avec une très bonne précision. Les concepts `Action`, `Actor`, `Condition`, `Modality`, `Reference` et `Time` obtiennent tous une précision égale ou supérieure à 0.80. Toutefois, le rappel moyen est plus faible, autour de 0.54, ce qui révèle une difficulté à détecter l'intégralité des termes présents dans les documents.
Cela est notamment dû au concept `Artifact`, qui affiche des performances significativement inférieures. Bien que la précision soit modérée (0.69), le rappel chute drastiquement à 0.16, entraînant un score F1 très faible de 0.26. Nos investigations montrent que ce résultat s'explique majoritairement par un fort déséquilibre de la distribution des instances dans le jeu de données d'entraînement, comme cela apparaît dans la Figure 7. Alors que la plupart des concepts sont répartis selon un ratio équilibré entre apprentissage et évaluation, le concept `Artifact` présente un déséquilibre marqué, avec seulement 74 occurrences dans l'ensemble d'entraînement contre 252 dans l'ensemble d'évaluation. Ce déséquilibre compromet la capacité du modèle à généraliser efficacement pour ce concept, ce qui explique les difficultés rencontrées. Pour remédier à cette situation, un réajustement des répartitions entre données d'apprentissage et d'évaluation serait nécessaire.
En résumé, le réentraînement du modèle CamemBERT a permis d'atteindre de bonnes performances, avec un score F1 moyen de 0.69 si l'on exclut le concept `Artifact`. Toutefois, les résultats obtenus soulignent clairement la dépendance du modèle à une quantité significative de données annotées, en particulier pour garantir une couverture adéquate. La constitution de tels jeux de données demeure une tâche exigeante en termes de temps et de compétences. Nous évaluons, dans la section suivante, la capacité des LLM à s'affranchir du corpus d'entraînement.

5.3 Résultats avec un LLM

Cette section présente les résultats obtenus à l'aide de l'ingénierie des requêtes et du LLM. Dans notre cas, nous utilisons GPT-4 pour l'extraction des termes juridiques. Les performances en précision, rappel et F1 sont présentées dans le Tableau 3.

	Concepts							
	Action	Actor	Artifact	Condition	Location	Modality	Reference	Time
Précision	0.65	0.67	0.58	0.76	0.53	0.54	0.71	0.81
Rappel	0.27	0.58	0.33	0.53	0.36	0.54	0.53	0.34
F1	0.38	0.63	0.42	0.63	0.42	0.54	0.61	0.48

TABLE 3 – Résultats de précision, rappel et F1 avec GPT-4 pour l'extraction de termes juridiques.

Les résultats obtenus révèlent une forte hétérogénéité selon les concepts. Le meilleur score de précision concerne le concept `Time` (0.81), ce qui illustre la capacité du modèle à identifier correctement les expressions temporelles. Des scores de précision élevés sont également observés pour les concepts `Condition` (0.76) et `Reference` (0.71). Sur le plan du rappel, le concept `Actor` atteint un score notable de 0.58, indiquant une bonne couverture des termes relevant de cette catégorie. En revanche, le concept `Action` affiche le rappel le plus faible (0.27), traduisant des difficultés à identifier de manière exhaustive les actions décrites dans les textes juridiques.
Ces résultats montrent une efficacité en termes de précision, mais soulignent aussi les limitations des LLM en matière de rappel. Le modèle GPT-4 sait identifier les termes de manière exacte lorsqu'il les reconnaît, mais il en omet un nombre significatif. Comme évoqué dans l'état de l'art, cette limitation a déjà été identifiée par des travaux précédents, révélant que les LLM ont tendance à annoter qu'une partie des expressions attendues.
Néanmoins, cette approche sans réentraînement offre des avantages notables, en particulier en termes de réduction du coût lié à l'annotation manuelle. En exploitant ses capacités fondationnelles, GPT-4 permet d'automatiser partiellement le processus d'extraction sans avoir recours à un corpus annoté dédié, ce qui facilite l'extension à d'autres domaines. Malgré des performances limitées sur certains concepts, cette approche réduit l'intervention humaine. Nous verrons dans la section suivante comment l'approche hybride permet d'en améliorer les résultats.

5.4 Résultats du système hybride

Les résultats de notre approche hybride, qui combine GPT-4, CamemBERT et un filtrage syntaxique, montrent une amélioration significative des performances globales pour l'extraction de termes juridiques. Le Tableau 4 présente les scores en précision, rappel et F1 obtenus pour chacun des concepts étudiés.

	Concepts							
	Action	Actor	Artifact	Condition	Location	Modality	Reference	Time
Précision	0.36	0.80	0.58	0.68	0.35	0.73	0.66	0.64
Rappel	0.78	0.52	0.54	0.73	0.47	0.57	0.75	0.81
F1	0.50	0.63	0.56	0.70	0.41	0.64	0.70	0.72

TABLE 4 – Résultats de précision, rappel et F1 avec notre approche hybride pour l'extraction de termes juridiques.

Le Tableau 5.4 met en évidence l'amélioration significative apportée par l'approche hybride par rapport à GPT-4. En combinant les capacités d'extraction initiale de GPT-4, le filtrage syntaxique et la distillation dans un CamemBERT, nous obtenons un gain de 7 à 24 points de pourcentage sur

Concept	CamemBERT	GPT-4	Hybride
Action	0.57	0.38	**0.50 (+12 %)**
Actor	0.70	0.63	0.63 (0 %)
Artifact	0.26	0.42	**0.56 (+14 %)**
Condition	0.82	0.63	**0.70 (+7 %)**
Location	0.59	**0.42 (+1 %)**	0.41
Modality	0.63	0.54	**0.64 (+10 %)**
Reference	0.76	0.61	**0.70 (+9 %)**
Time	0.74	0.48	**0.72 (+24 %)**

TABLE 5 – Comparaison des performances en F1 pour les trois approches : CamemBERT, GPT-4 et système hybride. En gras, les meilleures performances entre GPT-4 et Hybride.

le score F1 selon les concepts. Ces améliorations sont particulièrement notables pour les concepts les plus complexes ou mal représentés tels que `Artifact`, `Reference` et `Time`.

L'utilisation de règles syntaxiques permet d'atténuer les erreurs d'extraction apparaissant avec les LLMs. Ce filtrage améliore sensiblement la qualité des annotations automatiques générées, qui sont ensuite utilisées pour réentraîner CamemBERT. Le modèle résultant bénéficie à la fois de la capacité générative des LLMs et de l'efficacité prédictive d'un classifieur BERT, tout en limitant l'implication des experts.

Il convient toutefois de noter que le système hybride n'égale pas les performances du modèle CamemBERT entraîné de manière supervisée (sur un jeu de données annoté manuellement par des experts). Ce constat rejoint les observations présentées dans des travaux similaires, tels que celui de Tang et al. [29], soulignant qu'un entraînement basé sur des annotations expertes produit de meilleurs résultats.

Malgré cela, notre analyse démontre que l'approche hybride constitue une solution efficace pour l'extraction de termes juridiques tout en réduisant la dépendance à une annotation experte. Elle ouvre ainsi la voie à des systèmes plus autonomes, adaptés au traitement de corpus juridiques de grande taille, tout en maintenant un bon compromis entre qualité, coût, et effort humain.

6 Conclusion

Notre article a proposé une approche visant à améliorer l'extraction de termes juridiques, tout en réduisant l'implication des experts. Nous avons évalué trois stratégies distinctes pour l'extraction de termes juridiques : le réentraînement d'un modèle CamemBERT, l'exploitation d'un grand modèle de langage (LLM) via l'ingénierie des requêtes, ainsi qu'une approche hybride combinant LLMs, méthodes à base de règles syntaxiques, et un modèle CamemBERT. L'évaluation de ces stratégies a été menée sur un même jeu de données issu de la législation luxembourgeoise.

Nous avons montré que le réentraînement du modèle CamemBERT permet d'obtenir des performances élevées, démontrant sa capacité à extraire efficacement des termes juridiques à partir de textes en français. Par ailleurs, nous avons exploré le potentiel de GPT-4 pour réduire la nécessité d'annotation experte, en soulignant le rôle central de l'ingénierie des requêtes dans la production de sorties structurées. Enfin, l'approche hybride, fondée sur la distillation des connaissances d'un LLM vers CamemBERT via un filtrage par règles, s'est montrée particulièrement prometteuse. Celle-ci améliore significativement les performances de GPT-4 seul, tout en limitant l'intervention des experts à la définition des concepts, de leurs définitions, et à l'élaboration des règles syntaxiques de filtrage.

L'approche hybride présente des atouts majeurs par rapport aux méthodes traditionnelles de l'état de l'art. En tirant parti de GPT-4 pour générer des données annotées (« amorçage » ou bootstrapping), la dépendance au travail d'annotation devient minimale, à condition que le domaine étudié soit bien couvert par les connaissances du LLM.

Cependant, il existe des limites à cette approche hybride. Premièrement, le filtrage syntaxique suppose un accès à un analyseur syntaxique fiable, qui peut varier selon la langue. Ainsi, les règles développées pour le français ne sont pas directement transposables à d'autres langues, comme l'anglais, et requièrent un nouveau travail d'expertise. Par ailleurs, le modèle CamemBERT est spécifiquement entraîné pour le français ; son utilisation sur des documents rédigés dans d'autres langues nécessiterait l'adoption de modèles adaptés, tels que BERT pour l'anglais.

Après l'extraction terminologique, les perspectives de ce travail visent à extraire les relations entre termes juridiques. Des améliorations de l'approche hybride sont également envisagées, en particulier sur le plan du filtrage syntaxique. Au lieu de s'en remettre exclusivement aux experts pour la formalisation des règles, des techniques d'apprentissage non supervisé pourraient être envisagées pour assister, voire automatiser, leur génération, réduisant ainsi davantage l'effort manuel requis.

Remerciements

Ce travail a bénéficié d'un accès aux ressources de calcul intensif de l'IDRIS dans le cadre de l'allocation 2024-AD011014922 accordée par GENCI.

Références

[1] Patrizio Bellan, Mauro Dragoni, and Chiara Ghidini. Extracting business process entities and relations from text using pre-trained language models and in-context learning. In *International Conference on Enterprise Design, Operations, and Computing*, pages 182–199. Springer, 2022.

[2] David Biesner, Rajkumar Ramamurthy, Robin Stenzel, Max Lübbering, Lars Patrick Hillebrand, Anna Ladi, Maren Pielka, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. Anonymization of german financial documents using neural network-based language models with contextual word representations. *International Journal of Data Science and Analytics*, pages 151–161, 2022.

[3] Julien Breton, Mokhtar Boumedyen Billami, Max Chevalier, and Trojahn Cassia. Leveraging seman-

tic model and llm for bootstrapping a legal entity extraction : An industrial use case. In *20th International Conference on Semantic Systems (SEMANTICS 2024)*, 2024. to appear.

[4] Julien **Breton**, Mokhtar Boumedyen Billami, Max Chevalier, and Cassia Trojahn. Empowering CamemBERT Legal Entity Extraction With LLM Boostrapping. In *Knowledge Engineering and Knowledge Management*, volume 15370, pages 86–101. Springer Nature Switzerland, Cham, 2025.

[5] Silvana Castano, Alfio Ferrara, Emanuela Furiosi, Stefano Montanelli, Sergio Picascia, Davide Riva, and Carolina Stefanetti. Enforcing legal information extraction through context-aware techniques : The ASKE approach. *Computer Law & Security Review*, 52 :105903, 2024.

[6] Yun Chen, Bo Xiao, Zhiqing Lin, Cheng Dai, Zuochao Li, and Liping Yan. Multi-label text classification with deep neural networks. In *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pages 409–413. IEEE, 2018.

[7] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1) :1418, 2024.

[8] Mauro Dragoni, Serena Villata, Williams Rizzi, and Guido Governatori. Combining NLP approaches for rule extraction from legal documents. In *1st Workshop on MIning and REasoning with Legal texts (MIREL 2016)*, 2016.

[9] Alex Dunn, John Dagdelen, N. Walker, Sanghoon Lee, Andrew S. Rosen, G. Ceder, Kristin A. Persson, and Anubhav Jain. Structured information extraction from complex scientific text with fine-tuned large language models. *ArXiv*, 2022.

[10] Luca Foppiano, Guillaume Lambard, Toshiyuki Amagasa, and Masashi Ishii. Mining experimental data from materials science literature with large language models : an evaluation study. *Science and Technology of Advanced Materials : Methods*, 4, 2024.

[11] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm : Knowledge distillation of large language models, 2024.

[12] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander J. Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step ! outperforming larger language models with less training data and smaller model sizes. *ArXiv*, 2023.

[13] Yan Hu, Xu Zuo, Yujia Zhou, Xueqing Peng, Jimin Huang, Vipina K Keloth, Vincent J Zhang, Ruey-Ling Weng, Qingyu Chen, Xiaoqian Jiang, et al. Information extraction from clinical notes : Are we ready to switch to large language models ? *arXiv preprint arXiv :2411.10020*, 2024.

[14] Tomoki Ito and Shun Nakagawa. Tender document analyzer with the combination of supervised learning and llm-based improver. In *Companion Proceedings of the ACM Web Conference 2024*, pages 995–998, 2024.

[15] John Lafferty, Andrew McCallum, Fernando Pereira, et al. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Icml*, page 3. Williamstown, MA, 2001.

[16] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. Finding the law : Enhancing statutory article retrieval via graph neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, page 2753–2768, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[17] Larry R Medsker, Lakhmi Jain, et al. Recurrent neural networks. *Design and Applications*, 5(64-67) :2, 2001.

[18] Behrang Mohit. Named entity recognition. In *Natural language processing of semitic languages*, pages 221–245. Springer, 2014.

[19] Vitor Oliveira, Gabriel Nogueira, Thiago Faleiros, and Ricardo Marcacini. Combining prompt-based language models and weak supervision for labeling named entity recognition on legal documents. *Artificial Intelligence and Law*, 2024.

[20] Kalyani Pakhale. Comprehensive overview of named entity recognition : Models, domain-specific applications and challenges. *arXiv preprint arXiv :2309.14084*, 2023.

[21] Riccardo Pozzi, Riccardo Rubini, Christian Bernasconi, and Matteo Palmonari. Named entity recognition and linking for entity extraction from italian civil judgements. In Roberto Basili, Domenico Lembo, Carla Limongelli, and Andrea Orlandini, editors, *AIxIA 2023 – Advances in Artificial Intelligence*, pages 187–201. Springer Nature Switzerland, 2023.

[22] P Lakshmi Prasanna and D Rajeswara Rao. Text classification using artificial neural networks. *International Journal of Engineering & Technology*, 7(1.1) :603–606, 2018.

[23] Philippe Sassier and Dominique Lansoy. *Ubu loi*. Arthème Fayard, France, 2008.

[24] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *ArXiv*, 2018.

[25] Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, Marcello Ceci, and John Dann. An automated framework for the extraction of semantic legal metadata from legal texts. *Empirical Software Engineering*, 26 :1–50, 2021.

[26] Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, and John Dann. Automated extraction of semantic legal metadata using natural language processing. *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 124–135, October 2018.

[27] Firdaus Solihin, Indra Budi, Rizal Fathoni Aji, and Edmon Makarim. Advancement of information extraction use in legal documents. *International Review of Law, Computers & Technology*, 35(3) :322–351, 2021.

[28] Omri Suissa, Avshalom Elmalech, and Maayan Zhitomirsky-Geffet. Text analysis using deep neural networks in digital humanities and information science. *Journal of the Association for Information Science and Technology*, 73(2) :268–287, 2022.

[29] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does synthetic data generation of llms help clinical text mining ? arxiv 2023. *arXiv preprint arXiv :2303.04360*, 2023.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] Karthik S Vedula, Annika Gupta, Akshay Swaminathan, Ivan Lopez, Suhana Bedi, and Nigam H Shah. Distilling large language models for efficient clinical information extraction. *arXiv preprint arXiv :2501.00031*, 2024.

[32] Zhili Wang, Yufan Wu, Pengbin Lei, and Cheng Peng. Named entity recognition method of brazilian legal text based on pre-training model. *Journal of Physics : Conference Series*, 1550 :032149, 2020.

[33] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv :2302.10205*, 2023.

[34] Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Empirical study of zero-shot ner with chatgpt. *arXiv preprint arXiv :2310.10035*, 2023.

[35] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER : Generalist model for named entity recognition using bidirectional transformer.

[36] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv :2303.18223*, 2023.

[37] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. UniversalNER : Targeted distillation from large language models for open named entity recognition.

Représentation des connaissances pour l'interrogation neuro-symbolique des jumeaux numériques de bâtiment

Stéphane Reynaud[1,2], Anthony Dumas[1], Ana Roxin[2]

[1] B27-AI, Recherche & Développement

[2] Université Bourgogne Europe, Laboratoire d'Informatique de Bourgogne (LIB) UR 7534

sreynaud@b27.fr, adumas@b27.fr, ana-maria.roxin@ube.fr

Résumé

Cette étude propose une approche pour améliorer l'accessibilité et l'interprétabilité des données des jumeaux numériques de bâtiment (« Digital Building Twins », DBT) à travers des requêtes en langage naturel. En utilisant des ontologies spécifiques et des techniques d'intelligence artificielle (IA), notamment l'IA neuro-symbolique, cette méthode facilite l'extraction rapide d'informations spécifiques sur les bâtiments, améliorant ainsi l'efficacité du processus de requêtage des informations des DBT.

Mots-clés

Jumeau numérique, bâtiment, représentation des connaissances, intelligence artificielle neuro-symbolique.

Abstract

This study presents an approach to enhance the accessibility and interpretability of Digital Building Twin (DBT) data through natural language queries. By leveraging domain-specific ontologies and artificial intelligence (AI) techniques, including neuro-symbolic AI, the method enables rapid extraction of specific building details, improving the efficiency of DBT information querying.

Keywords

Digital twin (DT), buildings, knowledge representation, neuro-symbolic artificial intelligence (NSAI).

1 Introduction

Les projets de construction font face à une complexité croissante, impliquant de multiples parties prenantes (architectes, ingénieurs, entreprises, etc.) et de vastes volumes de données [1]. Pour traiter cette complexité, le secteur de l'architecture, de l'ingénierie et de la construction (« Architecture, Engineering, Construction », AEC) a réalisé d'importants investissements dans la numérisation. Plusieurs standards, outils et formats ont émergé, notamment autour de la notion de jumeau numérique de bâtiment (« Digital Building Twin », DBT). Un DBT vise à modéliser de manière numérique et en temps réel un bâtiment, intégrant des données géométriques, sémantiques, et contextuelles (historique, simulation, prédiction, etc.) [2].

Cependant, la grande diversité des sources de données et le besoin d'interopérabilité entre différentes normes amènent des défis majeurs. La norme ISO 16739 « Industry Foundation Classes » (IFC) représente le seul format ouvert pour les maquettes numériques [3], mais demeure complexe et lourde à interroger directement [4]. Par ailleurs, la capacité d'extraire rapidement et avec précision des informations du DBT (par exemple, les dimensions d'un type particulier de porte ou la liste des murs extérieurs à un étage donné) demeure délicate : l'utilisateur doit naviguer dans des arbres de classes ou employer des logiciels experts ou propriétaires. L'objectif de notre étude est donc de faciliter l'interrogation des jumeaux numériques de bâtiments grâce à l'intelligence artificielle (IA), et plus particulièrement au questionnement en langage naturel (« Natural Language Processing », NLP). Nous proposons une approche neuro-symbolique : d'une part, la connaissance est représentée formellement en utilisant des ontologies (pour décrire le bâtiment, ses éléments, ses propriétés, etc.) ; d'autre part, des techniques de réseaux neuronaux sont employer afin d'interpréter et de formuler des requêtes.
Cet article a été publié dans le second atelier sur l'intelligence artificielle pour les jumeaux numériques et les applications cyber-physiques (AI4DTCP) en conjonction avec la 33e conférence internationale conjointe sur l'intelligence artificielle (IJCAI 2024) [5].

2 Contexte scientifique

Cette section expose les définitions et les références nécessaires pour comprendre les composantes de notre approche.

2.1 Modélisation des connaissances

Le Web sémantique vise à rendre l'information accessible et exploitable de façon automatique, en annotant les données via des standards du W3C[1]. Une ontologie décrit un domaine en termes de classes, propriétés et axiomes logiques, permettant de faire du raisonnement automatisé. Dans ce cadre, SPARQL (« SPARQL Protocol and RDF Query Language ») est le langage de requête adapté pour extraire ou modifier des entités RDF (« Resource Description Framework »). Les graphes de connaissances (« Knowledge Graphs », KG) résultent de l'agrégation de connaissances hétérogènes, exprimées au travers de différentes ontologies,

[1] https://www.w3.org/

et ce à travers des triplets RDF (sujet - prédicat - objet). Ils peuvent prendre en compte des alignements multiples entre ontologies pour fusionner différents silos de connaissances [6].

2.2 OpenBIM

L'OpenBIM définit un processus collaboratif, promu par buildingSMART International, visant l'interopérabilité des données de construction à travers des normes ouvertes [5]. Le cœur d'OpenBIM repose sur le format IFC, devenu la référence pour décrire numériquement un bâtiment tout au long de son cycle de vie [3]. Malgré son adoption large, le modèle IFC, dans ses différentes mises en œuvre, a montré des limites en matière de requêtage et d'analyse [4]. La raison majeure réside dans la richesse descriptive et la complexité des structures IFC : il s'agit surtout de maximiser les capacités d'intégration du format pour les éditeurs de logiciels, ce qui, dans la pratique, engendre des modèles volumineux et fortement hiérarchisés [7]. De plus, si l'IFC est souvent promu comme vecteur d'interopérabilité, son utilisation concrète pour agréger et traiter des données provenant de multiples sources reste délicate, ne serait-ce par le fait que les identifiants des classes IFC ne sont uniques qu'au sein d'un même fichier [8].

2.3 Interopérabilité

3 niveaux d'interopérabilité peuvent être distingués dans les normes actuelles [9]:

1. Physique : être capable de manipuler, lire, stocker un flux de données (problème résolu par les protocoles de la couche « Transport » comme TCP/IP ou encore HTTP).
2. Syntaxique : pouvoir échanger des données structurées (résolue également au travers l'utilisation de langages basés sur XML).
3. Sémantique : partager l'interprétation des concepts au travers l'utilisation de vocabulaires (RDF) ou ontologies (OWL) communes.

Les DBT, qui doivent intégrer des données venant de multiples systèmes (capteurs, bases de documents variées, etc.), ont besoin d'un fort niveau d'interopérabilité sémantique. Ceci implique souvent l'alignement de plusieurs ontologies ou la construction de ponts de correspondance formels [8].

2.4 Traitement du langage naturel

Le NLP permet à des algorithmes de comprendre, analyser et générer du texte en langage naturel [10]. Parmi les évolutions récentes, on trouve les modèles « Transformer », qui ont supplanté les approches existantes dans de nombreux cas. Ces modèles se distinguent par leur mécanisme d'attention, qui permet de gérer efficacement les dépendances à longue distance et le contexte [11].
Dans le cadre d'une interface de question-réponse sur un DBT, le NLP est mobilisé à deux niveaux [12]:

— Compréhension du langage naturel (« Natural Language Understanding », NLU) pour analyser la phrase de l'utilisateur (détection des mots-clés, entités, intentions, etc.).
— Génération en langage naturel (« Natural Language Generation », NLG) pour produire une réponse compréhensible.

Nous nous concentrons sur la partie NLU, tout en proposant un module de génération de texte pour répondre à l'utilisateur.

3 Travaux connexes

Cette section présente les principales approches et recherches liées à notre étude.

3.1 Modélisation des connaissances

Le développement de l'ontologie ifcOWL a marqué un jalon pour la sémantisation du BIM [13]. Plusieurs équipes ont ensuite proposé des simplifications ou extensions pour mieux gérer son implémentation ou son requêtage. Certaines approches restreignent le domaine à quelques classes clés pour diminuer la charge d'inférences et d'autres intègrent des ontologies génériques ou métiers [4], [14], [15], [16].

3.2 Approches pour aligner les ontologies

Comme déjà évoqué, dans le domaine de l'AEC, la multiplicité des vocabulaires complique la mise en commun des données. Pour y remédier, des correspondances sont établies entre les entités, basées sur leur sens ou leur structure. Les travaux distinguent 4 méthodes principales [17] :

1. L'analyse terminologique (similitudes lexicales entre labels),
2. L'analyse structurelle (comparaison des hiérarchies et restrictions),
3. L'analyse basée sur les instances (statistiques, données réelles).
4. L'analyse sémantique (sens et contexte des entités)

L'alignement d'ontologies demeure un véritable défi pour la communauté scientifique [18].

3.3 Systèmes de question-réponse

Dans un système de question-réponse sur KB (« Knowledge Base Question Answering », KBQA) deux grandes stratégies existent :

— L'analyse sémantique, qui convertit la question en une forme logique intermédiaire (ex. un ensemble de triplets) [19];
— La recherche d'information, qui utilise des heuristiques ou des vecteurs pour parcourir la base de connaissances à la recherche de sous-ensembles d'entités pertinentes [20].

Les recherches récentes se tournent vers des méthodes hybrides combinant ces 2 approches, comme le raisonnement neuro-symbolique, où l'on combine la puissance des réseaux neuronaux pour reconnaître la sémantique du texte et la logique symbolique d'une ontologie pour le raisonnement exact [21].
Puisque l'ontologie ifcOWL existante ne fournit pas de descriptions ou de termes en langage naturel pour ses éléments, l'ontologie INLE [22] propose d'enrichir ifcOWL de représentations textuelles (avec synonymes, variations, etc.). C'est sur cette dernière que s'appuie notre approche.

3.4 Interopérabilité

Selon la norme ISO 11354-1:2011 [23], plusieurs approches existent pour l'interopérabilité des modèles :

- Intégration : tout décrire dans une ontologie harmonisée ;
- Unification : utiliser un méta-modèle commun comme pivot ;
- Fédération : laisser chaque source autonome en définissant des correspondances et/ou des ponts ponctuels.

Toutes ces stratégies peuvent permettre aux différentes parties prenantes de partager des données de manière cohérente, malgré des formats et des structures variés. Si l'on considère des ontologies comme des modèles de la connaissance d'un domaine, nous pouvons exploiter une de ces stratégies pour les rendre interopérables.

4 Notre approche

Notre approche s'appuie sur quatre piliers :

1. ifcOWL pour la structure formelle du bâtiment ;
2. Notre ontologie métier pour étendre et préciser certains sous-domaines (bois, menuiserie, HVAC, etc.) et associer des propriétés simplifiées ;
3. INLE pour gérer les expressions linguistiques (noms, synonymes, hyperonymes) des classes IFC/OB27AI ;
4. Un moteur d'analyse NLP, capable d'analyser la phrase en entrée, d'identifier les entités et relations, puis de les associer aux concepts ontologiques.

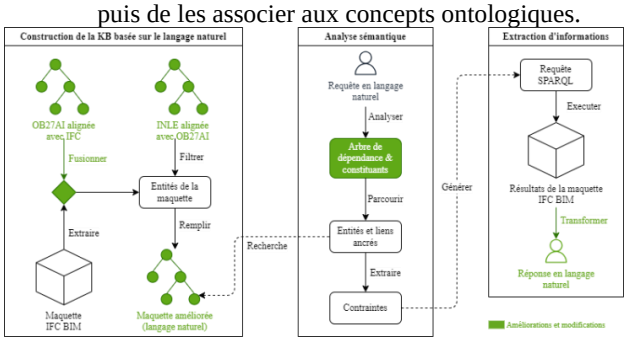

Figure 1 : Notre nouvelle approche comparée à sa version précédente [24]

En combinant ces éléments, notre système (cf. Figure 1) décompose la requête en langage naturel en sous-ensembles symboliques (triplets RDF), teste leur validité par rapport à la base de connaissance IFC et fusionne les résultats sous forme de requête SPARQL finale.

4.1 Modélisation des connaissances

Afin de pallier les limites de ifcOWL, et de mieux répondre aux besoins métier, une ontologie nommée OB27AI a été conçue selon une méthode éprouvée en sept étapes [25]. Les experts de chaque domaine (charpente bois, électricité, etc.) définissent des taxonomies de concepts. Les classes ainsi créées (ex. « PanneauPorte », « CadreFenêtre », etc.) conservent des noms en langage naturel (via la classe *b27:Nom*), permettant d'y associer des champs de synonymie (ex. « ouvrant », « vantail »). Des propriétés simplifiées (ex. *b27:aPourLargeur*, *b27:aMateriau*) y sont ajoutés, dont le comportement simplifie le raisonnement par rapport aux classes ou propriétés complexes IFC.

L'ontologie obtenue est plus fine sur certains objets, tout en évitant la surcharge axiomatique.

Pour faire en sorte que l'ontologie OB27AI fonctionne de pair avec ifcOWL et INLE, un alignement est réalisé, en définissant pour chaque classe source (*b27:Porte*, *b27:Poutre*, etc.) un lien logique (*rdfs:subClassOf*) vers la classe IFC correspondante (*ifc:IfcDoor*, *ifc:IfcBeam*). Les propriétés sont également reliées si nécessaire. Une partie de l'ontologie est consacrée aux noms en langage naturel (ex. vantail de porte, feuillure) avec leurs scores de similarité. Ainsi, puisque les classes IFC portent un lien dans INLE (*inle:isnlnameof*), les éléments de langage naturel (*inle:ifcdoor1*) sont associés par inférence avec la classe OB27AI correspondante (*b27:Porte*).

Ce travail d'alignement procure un double avantage :

- Hériter des propriétés et contraintes déjà présentes dans ifcOWL (par exemple, le fait qu'un *ifc:IfcDoor* puisse être lié à un *ifc:IfcDoorPanelProperties*).
- Enrichir la base IFC avec de nouvelles classes et propriétés plus granulaires, facilitant la requête sur des objets rarement nommés explicitement (ex. cadre de portes).

4.2 Interrogation en langage naturel

Pour interroger la base de connaissance IFC du DBT via des questions en langage naturel, nous employons une approche neuro-symbolique [21].

Tout d'abord, l'analyse sémantique repose sur un modèle d'IA statistique. Il permet de générer un arbre de dépendances et un arbre de constituants à partir de la question en entrée, dans le but d'en extraire la structure syntaxique et sémantique.

Ensuite, la représentation symbolique de chaque entité ou contrainte est détectée (ex. « mur extérieur », « hauteur > 3m ») et confrontée à l'ontologie OB27AI et aux ontologies ifcOWL/INLE, permettant de déterminer les classes et propriétés correspondantes dans la base de connaissances.

Ces éléments extraits sont combinés pour former progressivement des fragments de requête SPARQL. Un mécanisme de filtrage et de validation écarte les branches non pertinentes, ce qui renforce la précision.

Enfin, après exécution de la requête globale SPARQL, la liste brute d'objets renvoyée est convertie en texte lisible. Nous utilisons des gabarits statiques et des modèles neuronaux pour reformuler les réponses en langage naturel.

Cette approche neuro-symbolique exploite la souplesse du NLP (pour comprendre la question) et la logique ontologique (pour cibler finement les données du jumeau numérique). Elle s'avère particulièrement efficace pour manipuler des requêtes complexes (comparaisons, filtres multiples), dont la recherche manuelle dans l'IFC s'avérait longue et fastidieuse dans notre précédente approche [24].

5 Implémentation

L'architecture de notre prototype repose sur :

- Un magasin de triplets pour stocker et traiter les ontologies.
- Des scripts exploitant des bibliothèques de NLP existantes et des modèles de type « Transformer » pour le traitement des requêtes.

De plus, nous avons implémenter un processus en sept étapes pour convertir une requête utilisateur en une requête SPARQL, interrogeant le DBT et renvoyant des réponses exploitables et intelligibles :

1. Analyser : usage de d'analyseurs neuronaux (type « Transformer ») pour dégager un arbre de dépendances (dépendances syntaxiques) et un arbre de constituants (analyse hiérarchique).
2. Parcourir : construction d'un arbre d'entités candidates, basé sur l'identification des mots-clés (p. ex. « porte », « extérieur », « second étage »).
3. Ancrer : ancrage de ces entités dans le graphe OB27AI/INLE (pour repérer la classe *b27:Porte* par ex.) et vérification de liens possibles (ex. *b27:aMateriau*).
4. Extraire : exécution de requêtes SPARQL partielles sur la base IFC (convertie en RDF) pour filtrer et valider les contraintes (ex. « largeur > 1m »).
5. Générer : compilation du tout en une requête SPARQL finale.
6. Exécuter : exécution sur la base IFC, récupérant la liste brute des objets répondant aux critères.
7. Transformer : conversion des résultats en une phrase lisible (par ex. « Il y a 2 portes extérieures dont la largeur est supérieure à 1 mètre : Door_6597 et Door_6702. »).

6 Résultats et évaluation

L'approche a été testée sur un ensemble de 22 questions complexes portant sur des éléments architecturaux et structurels. Comparée à la version précédente du système [24], les résultats montrent :

- Un gain de 18,6 points en rappel (94,4% vs. 75,8%)
- Un score F1 amélioré (+10 points, atteignant 92,7%)
- Une réduction du temps de préparation des données de 44% grâce à l'utilisation d'un magasin de triplets

L'augmentation importante du rappel résulte notamment d'une meilleure détection des concepts et du chaînage des relations, rendue possible par notre ontologie OB27AI et un analyseur sémantique plus robuste.

En contrepartie, le temps total pour traiter une question a doublé (environ 2 secondes au lieu d'une) à cause de l'étape d'ancrage plus exhaustive.

7 Conclusion et perspectives

En combinant une ontologie dédiée (OB27AI) et une approche neuro-symbolique, il est possible d'interroger rapidement et efficacement un jumeau numérique de bâtiment en langage naturel. Les résultats obtenus (amélioration du rappel, forte réduction du temps de recherche pour l'utilisateur) valident l'approche. Les principaux points à retenir sont :

- L'alignement avec ifcOWL et INLE est un levier majeur pour associer la logique d'IFC et le langage naturel ;
- Les modèles neuronaux gèrent la variabilité lexicale et syntaxique, tandis que l'ontologie assure la cohérence sémantique et la précision des requêtes SPARQL ;
- Le coût computationnel est maîtrisé (quelques secondes par question), bien qu'il augmente avec la richesse de l'ontologie.

Pour la suite, plusieurs axes d'amélioration sont envisagés :

- Optimisation du processus d'ancrage des entités ;
- Réduction du temps de traitement en réduisant le nombre de modèles de NLP chargés ;
- Intégration d'un mécanisme de gestion du contexte conversationnel, permettant des requêtes enchaînées et plus complexes.

Ce travail ouvre ainsi la voie à une interaction plus fluide et efficace entre les utilisateurs et les maquettes numériques de bâtiments.

Remerciements

Nous tenons à exprimer notre gratitude à B27-AI pour ses contributions financières et matérielles à cette étude, et à l'Agence Nationale de la Recherche et de la Technologie (ANRT) pour sa subvention CIFRE.

8 Références

[1] A. Roxin, W. Abdou, et W. Derigent, « Interoperable Digital Building Twins Through Communicating Materials and Semantic BIM », *SN COMPUT. SCI.*, vol. 3, nº 1, p. 23, oct. 2021, doi: 10.1007/s42979-021-00860-w.

[2] O. A. Al-Mufti, O. A. Al-Isawi, L. H. Amirah, et C. Ghenai, « Digital Twinning and ANN-based Forecasting Model for Building Energy Consumption », in *2023 Advances in Science and Engineering Technology International Conferences (ASET)*, févr. 2023, p. 1-8. doi: 10.1109/ASET56582.2023.10180899.

[3] ISO 16739-1, *Classes IFC (Industry Foundation Classes) pour le partage des données dans le secteur de la construction et de la gestion de patrimoine*, 2024. [En ligne]. Disponible sur: https://www.iso.org/fr/standard/84123.html

[4] T. M. de Farias, A. Roxin, et C. Nicolle, « A rule-based methodology to extract building model views », *Automation in Construction*, vol. 92, p. 214-229, août 2018, doi: 10.1016/j.autcon.2018.03.035.

[5] S. Reynaud, A. Dumas, et A. Roxin, « Knowledge representation for neuro-symbolic digital building twin querying », in *The Second Workshop on AI for Digital Twins and Cyber-Physical Applications*, in CEUR Workshop Proceedings, vol. 3807. Jeju - South Korea, August: CEUR, août 2024, p. 48-75.

[6] A. Stellato, « Dictionary, Thesaurus or Ontology? Disentangling Our Choices in the Semantic Web

Jungle », *Journal of Integrative Agriculture*, vol. 11, nº 5, p. 710-719, mai 2012, doi: 10.1016/S2095-3119(12)60060-4.

[7] P. Pauwels *et al.*, « A semantic rule checking environment for building performance checking », *Automation in Construction*, vol. 20, nº 5, p. 506-518, août 2011, doi: 10.1016/j.autcon.2010.11.017.

[8] C. Zhang, J. Beetz, et de Vries, « BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data », *Semantic Web*, p. 1-27, août 2018, doi: 10.3233/SW-180297.

[9] H. Kubicek, R. Cimander, et H. J. Scholl, « Layers of Interoperability », in *Organizational Interoperability in E-Government: Lessons from 77 European Good-Practice Cases*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, p. 85-96. doi: 10.1007/978-3-642-22502-4_7.

[10] S. J. Russell et P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, 2021.

[11] A. Vaswani *et al.*, « Attention is All you Need », in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et R. Garnett, Éd., Curran Associates, Inc., 2017. [En ligne]. Disponible sur: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[12] D. Khurana, A. Koli, K. Khatter, et S. Singh, « Natural language processing: state of the art, current trends and challenges », *Multimed Tools Appl*, vol. 82, nº 3, p. 3713-3744, janv. 2023, doi: 10.1007/s11042-022-13428-4.

[13] P. Pauwels et W. Terkaj, « EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology », *Automation in Construction*, vol. 63, p. 100-133, mars 2016, doi: 10.1016/j.autcon.2015.12.003.

[14] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, et R. Klein, « The IFC to linked building data converter - current status », présenté à LDAC, 2018. Consulté le: 2 mai 2024. [En ligne]. Disponible sur: https://www.semanticscholar.org/paper/The-IFC-to-linked-building-data-converter-current-Bonduel-Oraskari/945040b77455d09b3f37f975fb18ab12624982d6

[15] P. Pauwels et A. Roxin, « SimpleBIM: From full ifcOWL graphs to simplified building graphs », in *Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM)*, 2017, p. 11-18.

[16] M. Bonduel, A. Wagner, P. Pauwels, M. Vergauwen, et R. Klein, « Including widespread geometry formats in semantic graphs using RDF literals », in *Proceedings of the 2019 European Conference for Computing in Construction*, European Council on Computing in Construction, 2019, p. 341-350. doi: 10.35490/ec3.2019.166.

[17] F. Ardjani, D. Bouchiha, et M. Malki, « Ontology-Alignment Techniques: Survey and Analysis », *IJMECS*, vol. 7, nº 11, p. 67-78, nov. 2015, doi: 10.5815/ijmecs.2015.11.08.

[18] B. Lima, D. Faria, et C. Pesquita, « Challenges of evaluating complex alignments. », in *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.*, 2021, p. 49-61. [En ligne]. Disponible sur: https://ceur-ws.org/Vol-3063/om2021_LTpaper5.pdf

[19] J. Berant, A. Chou, R. Frostig, et P. Liang, « Semantic Parsing on Freebase from Question-Answer Pairs », in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, et S. Bethard, Éd., Seattle, Washington, USA: Association for Computational Linguistics, oct. 2013, p. 1533-1544. Consulté le: 18 avril 2024. [En ligne]. Disponible sur: https://aclanthology.org/D13-1160

[20] A. Bordes, S. Chopra, et J. Weston, « Question Answering with Subgraph Embeddings », in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, et W. Daelemans, Éd., Doha, Qatar: Association for Computational Linguistics, oct. 2014, p. 615-620. doi: 10.3115/v1/D14-1067.

[21] J. Zhang, B. Chen, L. Zhang, X. Ke, et H. Ding, « Neural, symbolic and neural-symbolic reasoning on knowledge graphs », *AI Open*, vol. 2, p. 14-35, 2021, doi: https://doi.org/10.1016/j.aiopen.2021.03.001.

[22] M. Yin, L. Tang, C. Webster, S. Xu, X. Li, et H. Ying, « An ontology-aided, natural language-based approach for multi-constraint BIM model querying ».

[23] ISO 11354-1, *Advanced automation technologies and their applications — Requirements for establishing manufacturing enterprise process interoperability — Part 1: Framework for enterprise interoperability*, 2011. Consulté le: 18 avril 2024. [En ligne]. Disponible sur: https://www.iso.org/obp/ui/en/#iso:std:iso:11354:-1:ed-1:v1:en

[24] S. Reynaud, A. Dumas, et A. Roxin, « Neuro-symbolic approach for querying BIM models », in *2023 17th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, nov. 2023, p. 62-69. doi: 10.1109/SITIS61268.2023.00019.

[25] N. Noy et D. Mcguinness, « Ontology Development 101: A Guide to Creating Your First Ontology », *Knowledge Systems Laboratory*, vol. 32, janv. 2001.

Génération et validation de données structurées

Victor Charpenay
Mines Saint-Étienne, LIMOS

victor.charpenay@emse.fr

Résumé

Les Transformers permettent de générer rapidement toute forme de données, structurées ou non. Si ces données sont toujours plausibles, elles sont parfois incohérentes. Or, la nature séquentielle du processus de génération amène à la pérpétuation ou l'accumulation d'erreurs facilement détectables lorsqu'un schéma de données est disponible. Cet article présente une méthode de génération et de validation conjointe reposant sur des grammaires à base de clauses de Horn (Definite Clause Grammars) issues de la programmation logique.

Mots-clés

Transformer, encodeur-décodeur, langue contrôlée, Prolog.

Abstract

Transformers can quickly generate any form of data, structured or not. These data are always plausible but sometimes inconsistent. The sequential nature of the generation procedure leads to the accumulation of errors that could easily be detected if a data schema is available. This article introduces a joint generation and validation method relying on Definite Clause Grammars, closely related to logic programming.

Keywords

Transformer, Encoder-Decoder, Controlled Language, Prolog.

1 Introduction

Les grands modèles de langage (*Large Language Models*, LLM) permettent un rapprochement sans précédent entre langue naturelle et données structurées, manipulables par une machine. Leur architecture est ajustée pour générer, par exemple, du code Python, des requêtes SQL ou des données au format JSON. Les LLM à la base des chatbots d'OpenAI, Google ou Mistral peuvent aussi transcrire des connaissances en un graphe RDF syntaxiquement correct, au format Turtle. Cependant, il est rare que ces modèles respectent des contraintes usuelles associées au langage, comme le respect d'un vocabulaire pré-existant pour la génération de prédicats.

On cherche par exemple à transformer le paragraphe suivant (tiré de la fiche Wikipédia de la ville de Saint-Étienne) en triplets RDF.

> Saint-Étienne, appelée « Sainté » en langage familier, renommée Armeville à la Révolution française, est une commune française située au sud-ouest de Lyon (60 km environ), au sud-est de Clermont-Ferrand (90 km environ) et dans le quart-sud-est de la France, en région Auvergne-Rhône-Alpes. C'est le chef-lieu du département de la Loire.

En réponse au prompt 'Traduis ce paragraphe en triplets RDF.', le chatbot de Mistral génère les triplets suivants :

```
@prefix ex: <http://example.org/> .

ex:Saint-Étienne a ex:Commune ;
    ex:nomFamilier "Sainté" ;
    ex:ancienNom "Armeville" ;
    ex:situéeÀ ex:Sud-Ouest_de_Lyon,
               ex:Sud-Est_de_Clermont-Ferrand,
               ex:Quart-Sud-Est_de_la_France ;
    ex:région ex:Auvergne-Rhône-Alpes ;
    ex:chefLieuDe ex:Département_de_la_Loire .

ex:Sud-Ouest_de_Lyon ex:distance "60 km" .
ex:Sud-Est_de_Clermont-Ferrand ex:distance "90 km" .
```

Le résultat paraît satisfaisant mais il ne réutilise aucun vocabulaire connu. Si l'on veut contraindre la génération en indiquant la liste des propriétés de schema.org comme seules propriétés autorisées, on obtient le résultat suivant :

```
<Saint-Étienne> a <Commune> ;
    <alternateName> "Sainté" ;
    <alternateName> "Armeville" ;
    <addressLocality> "Saint-Étienne" ;
    <addressRegion> "Auvergne-Rhône-Alpes" ;
    <addressCountry> "France" ;
    <geo> <Point> .
<Point> <latitude> 45.4397 ;
    <longitude> 4.3871 .
<Saint-Étienne> <distance> <Distance_Lyon> .
<Distance_Lyon> <value> 60 ;
    <unitCode> "km" .
<Saint-Étienne>
    <distance> <Distance_Clermont-Ferrand> .
<Distance_Clermont-Ferrand> <value> 90 ;
    <unitCode> "km" .
<Saint-Étienne> <isPartOf> <Loire> .
<Loire> a <Department> .
<Saint-Étienne> <isPartOf> <France> .
<France> a <Country> .
```

Le prompt utilisé dans ce cas était 'Traduis ce paragraphe en triplets RDF en utilisant uniquement les propriétés suivantes : *liste des propriétés de schema.org* [1].'. Ce graphe

1. https://schema.org/version/latest#propaz

obtenu sous contrainte est de nettement moins bonne qualité. Il ajoute des informations qui ne sont pas dans le texte original (coordonnées géographiques) et ne respecte pas la sémantique des propriétés de schema.org. La propriété `distance`, par exemple, doit être suivie d'un littéral associant un nombre à une unité de mesure (comme `"60 km"`). Il est relativement facile de vérifier ces contraintes a posteriori, une fois les triplets disponibles, mais étant donné la structure formelle de schema.org, la génération par le LLM pourrait aussi être guidée par une validation à la volée des triplets générés, permettant au LLM de générer des données structurées de meilleure qualité. C'est ce que propose cet article.

On s'intéresse en fait à n'importe quel modèle d'apprentissage générant une séquence de symboles (*tokens*) à partir d'un autre ensemble de symboles ; des modèles désignés comme *sequence-to-sequence*, principalement basés sur l'architecture Transformer [18]. Les symboles d'entrée peuvent être du texte, comme dans l'exemple ci-dessus mais ils peuvent aussi être la discrétisation d'une série temporelle ou d'un signal audio. Les modèles de transcription (*speech-to-text*) comme Whisper [13] peuvent par exemple être utilisés dans un contexte précis, dans lequel la langue est contrôlée, comme lors de la transcription de commandes vocales. Les résultats présentés en fin d'article montrent que la validation à la volée de la transcription textuelle d'une commande permet de diminuer la taille du modèle génératif, sans en réduire les performances.

Les modèles génératifs produisant des séquences sont itératifs : ils produisent les symboles l'un après l'autre. La méthode proposée ici est simple, elle consiste à valider chaque symbole généré selon une grammaire pré-définie et à appliquer un algorithme de retour sur trace (*backtracking*) lorsqu'un symbole n'est pas accepté par la grammaire. La section suivante (section 2) présente un formalisme de grammaire lui-même construit sur l'idée de retour sur trace : les grammaires à base de clauses de Horn. La section 3 présente ensuite les détails de l'approche de génération et validation conjointe, symbole par symbole. La section 4 donne les résultats d'une évaluation de l'approche dans deux cas de figure : la transcription de commandes simples et la reformulation de descriptions issues de Wikipédia dans une langue contrôlée. Un rapide aperçu de l'état de l'art sur la génération contrainte termine l'article (section 5).

2 Schémas et grammaires

Toute donnée structurée s'exprime dans un langage formalisé par une grammaire. Lorsqu'un schéma est disponible pour cette donnée, on peut y associer une autre grammaire plus contraignante. On s'intéresse donc ici à la spécification et l'utilisation de grammaires pour la génération séquentielle de texte.

Il existe de nombreux formalismes pour spécifier une grammaire. Le plus connu est certainement la forme de Backus-Naur mais le formalisme qui nous intéresse ici est celui qui s'inscrit dans l'histoire de Prolog : les grammaires à base de clauses de Horn (*Definite Clause Grammars*, DCG). Dans leur ouvrage de référence The Art of Prolog, Sterling et Shapiro en donne l'exemple suivant [17, p. 257] :

```
% grammar rules

sentence --> noun_phrase, verb_phrase.

noun_phrase --> determiner, noun_phrase2.
noun_phrase --> noun_phrase2.

noun_phrase2 --> adjective, noun_phrase.
noun_phrase2 --> noun.

verb_phrase --> verb.
verb_phrase --> verb, noun_phrase.

% vocabulary

determiner --> [the]; [a].
adjective --> [decorated].
noun --> [pieplate]; [surprise].
verb --> [contains].
```

Les grammaires DCG ont leur propre syntaxe mais elles peuvent être traduites simplement en Prolog. La première règle de cette grammaire correspond par exemple à la règle Prolog suivante :

```
sentence(S, S0) :-
  noun_phrase(S, S1),
  verb_phrase(S1, S0).
```

où les deux arguments de chaque prédicat sont des listes. La seconde liste est toujours le suffixe de la première, de façon à ce que les deux ensemble représentent une liste par différence, un concept couramment utilisé en programmation logique.

Une fois la grammaire transformée en programme logique, Prolog peut être utilisé pour la validation d'une séquence de symboles. Le fait suivant, qui inclut une phrase complète, sera vrai car la phrase est reconnue par la grammaire :

```
sentence(
  [the, pieplate, contains, a, surprise],
  []
).
```

et toute phrase qui n'est pas dans le langage conduirait à un echec de la résolution Prolog. La résolution Prolog permet aussi de décider si une séquence donnée est le début d'une phrase reconnue par la grammaire, comme ci-dessous :

```
sentence(
  [the, pieplate | Tail],
  []
).
```

On peut ainsi utiliser Prolog pour générer l'ensemble des phrases du langage bien que dans cet exemple, la résolution ne terminerait pas du fait de la récursivité des règles pour `noun_phrase` et `noun_phrase2`. L'exemple est représentatif de nombreuses grammaires qui reconnaissent un nombre infini de phrases — celles formalisant les langues naturelles, par exemple.

Les modèles de génération *sequence-to-sequence* par Transformer peuvent eux aussi générer des phrases d'un langage-cible, avec une probabilité associée, sans pour autant garantir que toutes les phrases générées appartiendront

bien au langage. En combinant la génération par Transformer avec une validation Prolog, on peut garantir que la génération est contrôlée et espérer que les phrases générées soient parmi les plus probables dans le langage-cible.

3 Détail de l'approche

L'approche présentée dans cet articule suppose l'existence d'un modèle *sequence-to-sequence* pré-entraîné et d'une grammaire DCG, que l'on peut respectivement décrire à travers les fonctions GENERATE et VALIDATE. Dans une configuration où une séquence d'entrée I et une séquence de sortie O sont utilisées par le modèle [2], la fonction GENERATE produit à partir de ces deux séquences un vecteur donnant la probabilité qu'un symbole t soit le suivant dans la séquence O, pour chacun des symboles du langage. La signature de la fonction est donc GENERATE : $T^l \times T^l \rightarrow \mathbb{R}^{|T|}$, où T est l'ensemble des symboles définis dans le langage et l est la longueur maximale d'une séquence. On suppose qu'une séquence incomplète est complétée par un symbole spécial (*padding*). La fonction VALIDATE retourne simplement vrai ou faux selon si une séquence quelconque est un préfixe valide. Sa signature est VALIDATE : $T^l \rightarrow \{\text{true}, \text{false}\}$.

La fonction GENERATE est itérative : elle ne génère qu'un seul symbole à la fois. Elle permet une validation à chaque symbole et un retour sur trace si la validation échoue. On peut donc combiner génération et validation avec retour sur trace, dans l'algorithme ci-dessous :

```
function GENERATE_AND_VALIDATE(I, O)
    P = GENERATE(I, O)
    tokens = FILTER(O, P)
    sorted = SORT(tokens, P)
    for all t ∈ sorted do
        O' = O + t
        if VALIDATE(O') then
            O'' = GENERATE_AND_VALIDATE(I, O')
            if O'' is complete then
                return O''
            end if
        end if
    end for
    return ∅
end function
```

La fonction GENERATE_AND_VALIDATE appelle d'abord GENERATE, puis trie les symboles du plus probables au moins probables. Dans cet ordre, chacun des symboles t est testé avec VALIDATE, qui vérifie que la séquence $O + t$ est reconnue par la grammaire. Si ce n'est pas le cas, on teste le symbole suivant, moins probable. Si, au contraire, la séquence est valide, la procédure continue récursivement, jusqu'à qu'un symbole spécial de fin de génération soit atteint ou que l'ensemble des symboles aient été testés. Dans ce dernier cas, un retour sur trace s'effectue.

Sans condition d'arrêt particulière, cet algorithme génère l'ensemble des phrases possibles du langage, en nombre potentiellement infini. Les algorithmes classiques de génération ont des conditions d'arrêt supplémentaires, comme une taille maximum pour la séquence de sortie O. Ici, on considère une forme générique de contrôle de la terminaison de l'algorithme, en y ajoutant une fonction FILTER. Cette fonction réduit l'ensemble des symboles à explorer sur la base de la séquence disponible O et des probabilités retournées par GENERATE. On peut par exemple définir $\text{FILTER}(O, P) = \emptyset$ lorsque la longueur de O atteint une valeur maximum. On peut aussi écarter les symboles avec une faible probabilité. Une pratique courante lors de la génération de séquences consiste à sélectionner le plus petit ensemble de symboles T_p dont la somme des probabilités dépasse la valeur p [6]. Une autre approche, évaluée dans la section suivante, consiste à écarter les symboles dont la probabilité ne serait pas supérieure à celle d'un tirage aléatoire :

$$\text{FILTER}(O, P) = \{t \mid p_t > \frac{1}{|T|}\} \quad (1)$$

Le principal défaut de l'approche par retour sur trace est qu'il nécessite de générer et valider l'ensemble des séquences possibles avant d'échouer complètement. Lorsque le modèle génératif est sous-optimal et génère des séquences quasi-aléatoires, le temps de calcul peut alors être prohibitif, voire infini si aucun filtrage des symboles n'est effectué et que la grammaire reconnaît une infinité de séquences. Ce semi-déterminisme est une propriété héritée de Prolog et de la méthode de résolution pour le raisonnement. Cependant, lorsque la distribution de probabilité de P est obtenu par application de la fonction SOFTMAX, un filtrage basé sur les valeurs de probabilité permet d'élaguer efficacement la majorité des branches à explorer.

L'enjeu du fitrage de symbole est donc plutôt de ne pas filtrer trop de symboles, pour permettre au modèle génératif de générer des séquences peu probables mais correctes du point de vue de la grammaire. L'évaluation de la section suivante montre que le filtrage défini par l'équation 1 est suffisamment permissif.

4 Évaluation

On évalue l'approche dans deux configurations. On évalue d'abord la transcription d'un signal audio en texte avec le modèle Whisper. Cette évaluation permettra de démontrer l'intérêt d'une génération contrôlée lorsque le modèle est imprécis. Whisper, en effet, est un modèle multilingue qui a été entraîné avec plus ou moins de paramètres sur des enregistrements fortement dominés par l'anglais. Si un enregistrement est en français, les variantes de Whisper avec de nombreux paramètres sont relativement précises mais elles sont trop coûteuses pour une transcription en temps réel sur un CPU. À l'inverse, les variantes avec peu de paramètres s'exécutent rapidement mais elles produisent souvent des termes qui ne sont même pas dans le vocabulaire français. Lorsque le langage est contrôlé, une validation symbole par symbole devrait permettre de générer des séquences correctes, même pour les variantes de Whisper avec peu de paramètres.

2. dans cette configuration, les modèles ont deux parties : en encodeur et un décodeur. On trouve aussi des modèles qui n'incluent qu'un décodeur, comme les modèles de la famille des *Generative Pretrained Transformers* (GPT) [14, 1], pour lesquels on définit alors $I = \emptyset$.

La deuxième configuration évaluée dans l'article est celle d'une génération de texte à partir de texte, dans le but de reformuler (et simplifier) le texte d'origine. On suppose ici que le texte-cible est dans une langue contrôlée qui sert d'intermédiaire à une représentation formelle, comme *Attempto Controlled English* (ACE), dont toutes les phrases valides peuvent être réécrites en axiomes OWL ou en règles SWRL [8]. La reformulation est une tâche que peuvent exécuter des modèles comme *Text-to-Text Transfer Transformer* (T5) [15] ou BART [9]. Ces modèles se distinguent des GPT —famille à laquelle appartiennent les modèles Mistral [7] — par le fait que la séquence d'entrée est traitée séparément de la séquence de sortie (dans une architecture de type encodeur-décodeur). De nombreuses expériences autour de ces modèles montrent qu'ils sont capables d'atteindre de bonnes performances avec peu de paramètres pour différentes tâches, dont la reformulation. Comme avec Whisper, on cherche donc à démontrer que la validation symbole par symbole permet une génération rapide sur CPU de données structurées (par le biais d'une langue contrôlée).

D'autres configurations auraient pu être testées, notamment avec un GPT, pour évaluer l'importance de passer par une langue contrôlée dans la génération. Des tests préliminaires suggèrent en effet que T5 et BART ne peuvent pas générer efficacement une structure JSON ou des triplets RDF, la probabilité d'occurrence de symboles comme '"' ou '<' étant trop faible quelque soit la séquence d'entrée. Un GPT, en revanche, est typiquement entraîné avec un corpus de texte plus diversifié et peut donc théoriquement générer directement des données structurées, au prix d'une génération plus lente. Ce type d'évaluation n'est pas considéré ici mais fera l'objet de travaux futurs.

4.1 Transcription

Pour évaluer Whisper, on prend comme cas d'usage l'envoi de commandes vocales en français à un robot mobile autonome. Les commandes envoyées sont par exemple 'à gauche' ou 'va vers la porte et arrête-toi'. On évalue deux variantes de Whisper, *tiny* et *small*, sur un jeu de données composé de 46 commandes vocales de 3s chacune, enregistrées par un micro d'ordinateur. Elles ont été enregistrées par quatre personnes différentes (trois femmes, un homme). Dans cette expérience, la grammaire utilisée est la plus simple possible. Elle ne reconnaît que quatre phrases : 'en avant', 'en arrière', 'à gauche', 'à droite'. Parmi les 46 commandes enregistrées, 16 sont conforme à cette grammaire.

Pour mesurer la qualité des différents modèles considérés, on interprète la tâche comme une tâche de classification, ce qui permet de calculer une précision et un rappel. Lorsque les deux variantes de Whisper génèrent du texte sans validation, le résultat est un vrai positif lorsque la séquence de sortie est exactement celle attendue, à la ponctuation près, et qu'elle est conforme à la grammaire. Lorsque la génération est accompagnée d'une validation, on considère une génération en deux étapes : si la séquence générée est reconnue par la grammaire, elle est prise telle quelle ; si elle n'est pas reconnue par la grammaire, est prise la séquence obtenue par génération gloutone (sans validation) par le modèle d'origine. Par exemple, lorsque la séquence attendue est 'avance tout droit', le modèle *tiny* avec validation génère 'à d' puis renvoie ∅. Par défaut, on prend alors la séquence générée par le modèle *tiny* seul (à savoir 'Avons-tu droit ?', qui est un vrai négatif).

	Params	**Précision**	**Rappel**	**F1**
tiny	39M	0.28	0.29	0.28
small	244M	0.70	0.69	0.69
tiny + `cmd`	39M	0.48	0.93	0.63

TABLE 1 – Évaluation de variantes de Whisper avec une grammaire simple de commandes vocales (`cmd`)

Le tableau de résultats (table 1) montre qu'une validation par Prolog bénéficie nettement au modèle *tiny*. Sans augmenter le nombre de paramètres du modèle et sans réentraînement, sa précision passe de 28% à 48% et son score F1 approche celui du modèle *small*, avec 6 fois moins de paramètres.

4.2 Reformulation

Pour évaluer les capacités de reformulation d'un modèle *sequence-to-sequence*, un autre protocole expérimental est nécessaire. La transcription est une opération fonctionnelle : il n'existe qu'une seule transcription possible pour un enregistrement audio donné, à la ponctuation près. Ce n'est pas le cas lorsqu'on reformule du texte. De nombreuses structures de phrases différentes peuvent avoir la même sémantique et donc être considérées comme des reformulations l'une de l'autre, même dans une langue contrôlée. La procédure d'évaluation la plus évidente consisterait donc à éliminer les variations de formulation en transformant en données structurées la séquence de sortie du modèle et à comparer sémantiquement les structures de données. Si une ontologie existe pour ces données, des équivalences sémantiques peuvent être inférées automatiquement. Le travail d'ingénierie pour aboutir à ce résultat étant conséquent, l'évaluation faite ici a est simplifiée. Elle ne présente que des résultats préliminaires.

Le langage ACE est un bon candidat pour la génération de données structurées. Au-delà du fait qu'il existe une transformation d'ACE vers les langages usuels du Web sémantique, l'outil d'analyse syntaxique développé par ses concepteurs se base en effet sur une grammaire DCG, qui pourrait être utilisée telle quelle pour la validation. Une évaluation complète avec ACE n'a pas encore été faite mais les résultats préliminaires présentés ci-dessous suggèrent que cette langue contrôlée est suffisamment proche de la langue naturelle pour l'envisager comme intermédiaire.

On choisit comme modèle le successeur de T5, FLAN-T5, entraîné sur une plus grande variété de tâches de génération (comme l'explication ou le raisonnement) [3]. Comme séquence d'entrée, cinq descriptions en anglais issues de Wikipédia ont été sélectionnées, parmi celles incluses dans

le jeu de données T-Rex. T-Rex a été construit dans le but d'entraîner des modèles de générations de triplets RDF à partir de texte [4]. Les cinq descriptions ont été choisies au hasard, en vérifiant qu'elle décrivent des entités de type différent : une personne, une organisation, un pays, un nom commun et un produit manufacturé. Ces descriptions ont été reformulées manuellement dans la langue ACE puis validées par son outil d'anayse syntaxique [3]. Elles sont reportées dans un tableau, table 2. Les descriptions d'origine incluses dans T-Rex, elles, ne respectent pas la syntaxe d'ACE.

À partir d'une séquence d'entrée issue d'une description Wikipédia et d'une séquence de sortie en ACE, on peut estimer la difficulté pour FLAN-T5 à générer la séquence de sortie. Pour ce faire, on collecte le rang de chaque symbole de la séquence selon les probabilités P estimées par le modèle. Il existe différentes variantes de FLAN-T5. On évalue ses deux versions les plus petites, *small* (80M paramètres) et *base* (250M paramètres). Étant donnée une séquence décrivant David Oliver Huffman, par exemple, FLAN-T5 *small* donne la probabilité la plus élevée au symbole 'David' pour commencer la génération (rang 0, voir table 2). À l'inverse, au dixième symbole (toujours 'David' mais en début de seconde phrase), le modèle donne une probabilité trop faible pour être traitée par l'algorithme selon l'équation 1 (rang -1). La séquence complète ne pourrait donc pas être générée. C'est aussi le cas pour la deuxième séquence, décrivant BMW, mais pas pour les trois suivantes (Kazakhstan, moine et Thalia). La variante FLAN-T5 *base*, elle, peut générer les cinq séquences.

Il apparaît aussi dans ces résultats préliminaires que le modèle génère difficilement certains termes, comme 'comes', 'year' ou 'transportation'. Si l'on compare à une génération directe en triplets RDF, ces termes correspondent à la génération d'un prédicat étant donné un sujet ($\langle s, ?, ? \rangle$) et d'un objet étant donné un sujet et un prédicat ($\langle s, p, ? \rangle$). Or, dans un graphe RDF, il existe rarement une solution unique à ces requêtes. Il n'est donc pas étonnant que le rang de ces symboles particulièrement soit élevé. En revanche, il est intéressant de noter que la méthode de filtrage décrite par l'équation 1 permet de conserver ces symboles importants pour la qualité des données générées, malgré leur rang élevé.

5 État de l'art

La génération de texte avec des modèles pré-entraînés peut se faire selon différentes approches. Étant donné que la fonction GENERATE renvoie une distribution de probabilités plutôt qu'un unique symbole, un choix doit être fait à chaque étape pour générer une séquence entière. L'approche la plus évidente est gloutone : elle consiste à sélectionner le symbole avec la plus haute probabilité. Il a cependant été démontré qu'elle ne permet pas de reproduire fidèlement la langue naturelle, beaucoup plus variée [6]. Pour pallier ce problème, il est possible de générer non pas une mais plusieurs séquences entières en parallèle et d'en choisir une après génération (par exemple, celle avec la plus grande probabilité cumulée). Pour cela, on peut choisir les k symboles les plus probables à chaque étape [5] ou les symboles dont la somme des probabilités dépasse un seuil p [6]. Ces différentes approches peuvent être combinées à un tirage aléatoire de symboles pour plus de diversité dans la génération. À titre d'exemple, la bibliothèque logicielle d'HuggingFace implémente huit stratégies différentes pour la génération de texte [4]. Elles ont toutes en commun de ne pas s'appuyer sur des connaissances ou un schéma connus a priori, contrairement à l'approche présentée dans cet article.

D'autres approches cherchant à contraindre syntaxiquement ou sémantiquement la sortie d'un modèle ont été proposées. Ces approches sont basées sur un ré-entraînement du modèle, soit en fixant le format des séquence d'entrée pendant l'apprentissage [10], soit en modifiant les paramètres d'apprentissage [16]. Il est aussi courant de régulariser la fonction de coût de l'apprentissage du modèle, approche qui a démontré son efficacité dans la génération directe de données structurées [11]. REBEL, un modèle de génération de triplets RDF à partir de texte, est le produit d'un ré-entraînement de BART sur des données issues de T-Rex [2]. Contrairement à ces approches par ré-entraînement, l'approche par génération et validation peut intégrer des modèles pré-entraînés « sur étagère ».

Pour finir, certains travaux dans le domaine de l'intégration neuro-symbolique ont un lien direct avec la question traitée dans cet article. DeepProbLog, notamment, est un outil qui combine modèles d'apprentissage et programmation logique [12]. Le programme Prolog suivant, incluant le prédicat `nn` issu de DeepProbLog, permet théoriquement de générer et valider des séquences symbole par symbole, avec retour de trace :

```
seq2seq(I, O, N) :- sentence(O, []).
seq2seq(I, O, N) :-
  % generation
  nn(I, O, Token, N),
  % validation
  append(O, [Token|Tail], Op),
  sentence(Op, []),
  % recursive call
  Np is N+1, seq2seq(I, Op, Np).
```

Le prédicat `nn` permet de faire appel à un modèle pré-entraîné, dont on suppose ici qu'il énumère les symboles possibles du plus probable au moins probable, étant donné une séquence d'entrée I et une séquence (partielle) de sortie O. Cependant, les détails d'implémentation de DeepProbLog font qu'un tel programme ne pourrait pas être exécuté en pratique, du fait que le moteur d'inférence de DeepProbLog cherchera à énumérer toutes les instanciations possibles de `seq2seq` avant de calculer leur probabilité — instanciations qui peuvent être en nombre infini.

3. https://github.com/Attempto/APE/

4. https://huggingface.co/docs/transformers/generation_strategies

Entité	**Rang** [*small*,*base*]
David Oliver Huffman	_David [0,1], _Oliver [0,0], _H [0,0], uff [0,0], man [0,0], _is [2,7], _an [3,0], _actor [2,1], . [1,5], _David [**-1**,19], _Oliver [0,0], _H [0,0], uff [0,0], man [0,0], _comes [**89**,**123**], _from [0,0], _the [0,1], _USA [31,314], . [0,0]
BMW	_BMW [4,0], _is [1,3], _ [0,0], a [0,0], _company [11,2], . [40,12], _BMW [**-1**,3], _produces [3,3], _luxury [10,11], _vehicles [0,0], . [0,1], _The [2,2], _year [**232**,17], _of [1,1], _creation [13,4], _of [2,2], _BMW [6,0], _is [0,0], _1916 [0,4], . [0,0]
Kazakhstan	_Kazakhstan [0,0], _is [0,0], _ [3,2], a [0,0], _country [0,0], . [26,17], _It [1,1], _is [2,0], _located [0,1], _in [0,0], _Central [0,0], _Asia [0,0], _and [0,0], _Eastern [0,0], _Europe [0,0], . [0,0]
Monk	_A [12,1], _mon [0,0], k [0,0], _is [0,0], _ [0,0], a [0,0], _religious [2,2], _person [0,1], . [1,1]
Thalia	_Th [175,518], alia [0,0], _is [1,16], _an [2,2], _aircraft [0,15], _for [15,24], _transportation [**93**,8], . [2,2], _It [1,1], _is [1,1], _used [1,6], _in [0,0], _Japan [5,1], _during [2,13], _World [0,0], _War [0,0], _II [0,0], . [0,0]

TABLE 2 – Rang de chaque symbole lors de la génération d'une séquence pré-définie dans la langue contrôlée ACE avec les variantes *small* et *base* du modèle FLAN-T5 (le caractère _ encode un espace)

6 Conclusion

Les résultats préliminaires présentés dans cet article sont prometteurs. Ils indiquent que la génération de texte avec des modèles *sequence-to-sequence* de petite taille, combinée à une validation symbole par symbole, permettrait d'obtenir des données structurées de qualité, une perspective intéressante pour le domaine de l'ingénierie des connaissances.

Il reste à confirmer ces résultats dans des expériences plus poussées, dans lesquelles seraient produits des triplets RDF selon un vocabulaire connu. Par ailleurs, l'approche devrait être comparée à une génération par GPT qui produirait directement des données structurées.

Références

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 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.

[2] Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL : relation extraction by end-to-end language generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics : EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2370–2381. Association for Computational Linguistics, 2021.

[3] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25 :70 :1–70 :53, 2024.

[4] Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. T-rex : A large scale alignment of natural language with knowledge base triples. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings*

of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA), 2018.

[5] Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1 : Long Papers*, pages 889–898. Association for Computational Linguistics, 2018.

[6] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[7] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023.

[8] Tobias Kuhn. The understandability of OWL statements in controlled english. *Semantic Web*, 4(1) :101–115, 2013.

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART : denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.

[10] Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. Commongen : A constrained text generation challenge for generative commonsense reasoning. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics : EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1823–1840. Association for Computational Linguistics, 2020.

[11] Tengfei Ma, Jie Chen, and Cao Xiao. Constrained generation of semantically valid graphs via regularizing variational autoencoders. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31 : Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7113–7124, 2018.

[12] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Neural probabilistic logic programming in deepproblog. *Artif. Intell.*, 298 :103504, 2021.

[13] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 2023.

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8) :9, 2019.

[15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21 :140 :1–140 :67, 2020.

[16] Lei Sha. Gradient-guided unsupervised lexically constrained text generation. 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 8692–8703. Association for Computational Linguistics, 2020.

[17] Leon Sterling and Ehud Y Shapiro. *The art of Prolog : advanced programming techniques*. MIT press, 1994.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

Prédiction de similarités entre vocabulaires : exemple de l'UMLS

Safaa Menad[1], Saïd Abdeddaïm[1], Lina F. Soualmia[1]

[1] Univ Rouen Normandie, INSA Rouen Normandie, Normandie Univ
LITIS UR 4108, FR-76000 Rouen, France

9 mai 2025

Résumé

Le métathésaurus de l'UMLS regroupe un grand nombre de terminologies médicales, souvent non alignées. Pour faciliter leur intégration, plusieurs méthodes ont été proposées, notamment des approches classiques fondées sur des correspondances lexicales. Toutefois, ces méthodes s'avèrent limitées pour identifier des relations sémantiques complexes entre concepts ne partageant pas nécessairement de termes évidents. Nous proposons une approche basée sur des transformeurs siamois pour améliorer l'alignement d'ontologies. Ces modèles permettent de mieux saisir les relations entre concepts biomédicaux. Nous exploitons l'apprentissage contrastif auto-supervisé sur des articles biomédicaux pour prédire les similarités entre concepts. Notre approche améliore la précision des alignements en intégrant le contexte sémantique, renforçant ainsi le métathésaurus UMLS en proposant de nouvelles relations. Nos résultats montrent le potentiel des modèles transformeurs pré-entraînés pour optimiser l'intégration des vocabulaires au sein de l'UMLS.

Mots-clés

Ontologies, Alignement, Métathésaurus UMLS, Ontologie biomédicale, Transformeurs, Réseau neuronal siamois, Similatité sémantique, Embeddings de phrases

Abstract

The UMLS Metathesaurus encompasses numerous medical terminologies, often misaligned. To facilitate their integration, several approaches have been proposed for aligning them, including traditional approaches based on lexical matching. However, these methods are limited in their ability to identify complex semantic relations between concepts that do not necessarily share obvious terms. We propose a siamese transformer-based approach to enhance ontology alignment, enabling a better understanding of biomedical concept relationships. We leverage self-supervised contrastive learning on biomedical literature to predict similarities between concepts. Our approach improves alignment accuracy by incorporating semantic context, strengthening the UMLS Metathesaurus by introducing new relationships. Our results highlight the potential of pretrained transformer models for optimizing ontology integration in the UMLS.

Keywords

Ontology, Alignment, UMLS Metathesaurus, Biomedical Ontology, Transformers, Siamese Neural Network, Semantic Similarity, Sentence Embeddings

1 Introduction

Les ontologies biomédicales catégorisent et organisent les informations cruciales pour la recherche et les applications biomédicales. Cependant, ces ontologies comprennent souvent des concepts hétérogènes mais qui sont sémantiquement liés. L'établissement de relations significatives entre ces concepts hétérogènes est d'une importance critique [15]. L'alignement d'ontologies constitue une solution à ce problème d'hétérogénéité sémantique en déterminant les correspondances entre concepts issus de différentes ontologies biomédicales.

La figure 1 illustre un exemple de relation entre deux concepts, *wound and injuries* et *traumatic injury*, issus de deux vocabulaires, *MeSH* et *CSP*, au sein de l'UMLS. L'objectif dans ce travail est d'aligner ces concepts et de proposer une relation d'équivalence entre eux.

Un alignement précis de ces ontologies est essentiel pour améliorer l'interopérabilité, faciliter l'intégration des données. Les méthodes d'alignement traditionnelles s'appuient souvent sur des comparaisons lexicales, mais celles-ci peinent à capturer les relations sémantiques complexes propres aux ontologies biomédicales. Avec l'évolution continue des données biomédicales, la terminologie biomédicale se caractérise par une complexité et une ambiguïté accrues, compliquant davantage le processus d'alignement.

Étant donné que la création manuelle d'alignements est chronophage et exigeante en ressources, notamment pour les grandes ontologies contenant des milliers de concepts, plusieurs méthodes d'alignement ont été développées pour générer automatiquement des correspondances ontologiques [3]. De plus, en raison de la faible expressivité sémantique de certaines ontologies, des ressources externes peuvent être exploitées pour enrichir le processus d'alignement.

Avec les avancées de l'apprentissage automatique, l'apprentissage profond a été proposé comme alternative [5]. Dans le domaine biomédical, certaines méthodes d'alignement d'ontologies basées sur l'apprentissage profond ont démontré leur potentiel pour améliorer l'interopérabilité

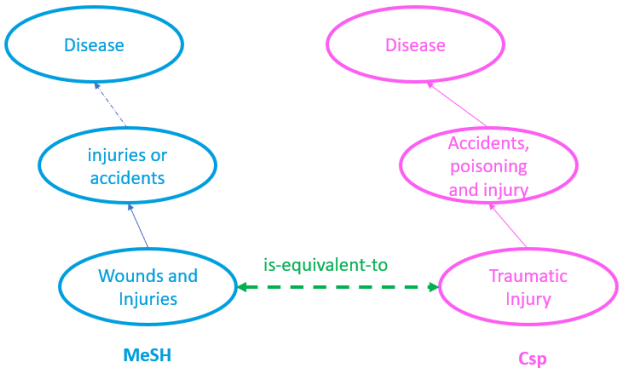

FIGURE 1 – Exemple d'une relation de similarité proposée.

entre ontologies [6, 7]. Cependant, ces méthodes reposent principalement sur des modèles de plongement lexical non contextuels, tels que Word2Vec. Les modèles de représentation du langage basés sur les transformeurs pré-entraînés, comme BERT [2], permettent d'apprendre des représentations contextuelles robustes et nécessitent généralement des ressources de calcul modérées pour l'ajustement. Bien que ces modèles soient performants sur de nombreuses tâches de traitement du langage naturel, leur application à l'alignement d'ontologies et à l'alignement de concepts reste encore peu explorée.

À travers ce travail, nous proposons une stratégie d'alignement plus efficace et plus fine que nous appliquons à l'UMLS.

Le *Unified Medical Language System* (UMLS) [1] est un système d'intégration de terminologies biomédicales contenant environ 200 vocabulaires sources, tels que MeSH, SNOMED CT, CHV et CPT. Il comprend également des vocabulaires bien définis (NCI, FMA et GO) qui sont souvent utilisés comme références pour la tâche d'alignement d'ontologies. Contrairement à ces ontologies bien structurées, tous les vocabulaires de l'UMLS ne sont pas nécessairement définis de manière formelle ou représentés sous forme d'ontologies. Ainsi, lorsque nous faisons référence au processus de construction du Métathésaurus, nous utilisons parfois la notion de vocabulaire plutôt que celle d'ontologie.

Le processus de construction et de maintenance du Métathésaurus UMLS est coûteux, chronophage et sujet aux erreurs, car il repose sur : i) un traitement lexical et sémantique pour déterminer les synonymies, et ii) l'expertise des éditeurs de l'UMLS pour valider ces synonymies.

Malgré les avancées des techniques d'alignement d'ontologies basées sur les réseaux neuronaux profonds, la mise en correspondance et l'intégration des terminologies restent des défis à l'échelle du Métathésaurus UMLS.

Face à ces défis, nous présentons une nouvelle approche exploitant les réseaux neuronaux siamois pour aligner les ontologies biomédicales. Les réseaux neuronaux siamois excellent dans la capture des nuances sémantiques, ce qui en fait un choix idéal pour les tâches d'alignement d'ontologies. Notre méthodologie, basée sur ces réseaux, vise à améliorer significativement la précision des alignements et à approfondir la compréhension des relations entre concepts biomédicaux.

Nos principales contributions dans cet article sont les suivantes : i) Nous introduisons nos modèles siamois que nous avons appliqués à la tâche d'alignement d'ontologies. ii) Nous démontrons empiriquement l'utilité de ces modèles pour l'alignement sémantique des entités issues d'ontologies biomédicales. À titre d'exemple, nous proposons d'aligner l'ensemble des vocabulaires de l'UMLS afin d'établir de nouvelles relations entre leurs terminologies distinctes.

2 Travaux antérieurs

Pour améliorer le processus d'intégration des vocabulaires dans l'UMLS, [12] ont développé une approche d'apprentissage supervisé permettant de suggérer des paires synonymes à l'échelle des vocabulaires sources de l'UMLS. Ils ont ensuite comparé leur approche à une méthode basée sur des règles, similaire au processus actuel de construction de l'UMLS. Les auteurs ont conçu et entraîné un réseau siamois afin de prédire si deux termes (atomes) de l'UMLS sont synonymes, puis ont utilisé la distance de Manhattan pour calculer la (dis)similarité entre les représentations finales issues du réseau siamois.

[10] ont proposé d'améliorer leur architecture précédente en y ajoutant une couche d'attention, affirmant que cette nouvelle approche accroît la précision.

Dans l'étude de [14], une approche visant à automatiser le mapping des terminologies externes à l'UMLS a été proposée. Elle combine deux techniques classiques d'alignement d'ontologies : une technique lexicale pour identifier les similarités textuelles, suivi d'une validation structurelle basée sur la compatibilité des concepts de haut niveau. Appliquée au thésaurus EMTREE, cette méthode a atteint une précision globale de 78%.

Dans [4], plusieurs modèles de langage tels que BERT, RoBERTa et GPT-2 ont été affinés et explorés pour l'alignement des vocabulaires de l'UMLS. La tâche d'alignement a été abordée soit comme un problème de classification, soit comme un problème de génération de texte. Les expériences ont démontré que l'intégration du contexte dans les entrées améliore significativement les performances.

Dans [11], plusieurs variantes de modèles d'apprentissage enrichis par le contexte (ConLMs) ont été développées en incorporant différents types d'informations contextuelles dans un modèle lexical (LexLM). Ces types de contexte sont représentés dans des graphes de connaissances enrichis (ConKGs) avec quatre variantes : ConSS, ConSG, ConHR et ConAll. Ces graphes ont ensuite été entraînés à l'aide de sept techniques d'apprentissage d'embeddings de graphes de connaissances. Les modèles ConLM ont été construits en concaténant les vecteurs d'embeddings des ConKGs avec les vecteurs lexicaux du LexLM. Les expé-

1. https://www.nlm.nih.gov/research/umls/index.html

riences ont montré une amélioration significative des performances des ConLMs par rapport aux LexLMs.
Les méthodes précédemment citées se concentrent généralement sur une portion limitée des concepts de l'UMLS ou reposent sur des architectures de type cross-encodeur (comme BERT) [13], qui sont coûteuses en temps de calcul et peu adaptées au traitement de grands volumes de données. Dans ce travail, nous proposons une approche générique, à la fois efficace et scalable, conçue pour traiter des données à grande échelle.

3 Approche

Dans ce travail, nous considérons l'alignement d'ontologies (*Ontology Matching*, OM) comme un problème de similarité. Nous proposons donc d'utiliser des réseaux de neurones siamois pour traiter les concepts ontologiques sous forme textuelle et calculer leur similarité en vue d'un alignement efficace. En exploitant la capacité inhérente des réseaux siamois à mesurer la similarité textuelle, notre approche transforme les concepts des ontologies biomédicales en représentations textuelles adaptées. Ces représentations sont ensuite utilisées pour calculer la proximité sémantique entre des paires de concepts issus d'ontologies distinctes.
Les sentence-transformeurs sont des modèles de langues développés pour la tâche de calcul de scores de similarité entre deux phrases. Ils utilisent des transformeurs pour des tâches liées aux paires de phrases, telles que la récupération d'information, la paraphrase de phrases, etc. Ces transformeurs reposent sur deux architectures : les cross-encoders, qui traitent la concaténation des paires de phrases, et les siamois bi-encoders, qui encodent chaque élément de la paire en vecteurs séparemment. La particularité des modèles siamois à bi-encodeurs réside dans leur manière de traiter les données en entrée. En calculant séparément les représentations vectorielles de chaque élément, ces représentations peuvent être pré-calculées et stockées, puis réutilisées efficacement. Cette propriété permet une optimisation significative du temps de traitement et facilite le passage à l'échelle pour de grands volumes de données.

3.1 Modèles utilisés

Les sentence transformeurs ont été initialement conçus pour transformer des phrases (de longueur similaire) en vecteurs. Dans notre approche, nous proposons de transformer les termes MeSH, les titres d'articles et les résumés de PubMed dans le même espace vectoriel en entraînant un modèle siamese transformer sur ces données. Notre objectif est d'assurer une correspondance dans cet espace vectoriel entre textes courts et longs. Ainsi, nous avons entraîné nos modèles en utilisant des paires d'entrées (titre, terme MeSH) et (résumé, terme MeSH).
Nous avons construit BioSTransformers [9] construit en utilisant un transformer pré-entraîné sur des données biomédicales, Nous nous sommes inspirés du modèle Sentence-BERT [13] en remplaçant BERT par d'autres transformeurs pré-entraînés sur des données biomédicales (bio-transformers). Nous utilisons dans ce travail la variante SBio_ClinicalBERT qui est basé sur le modèle

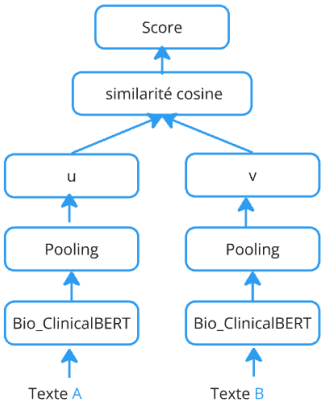

FIGURE 2 – Architecture du modèle.

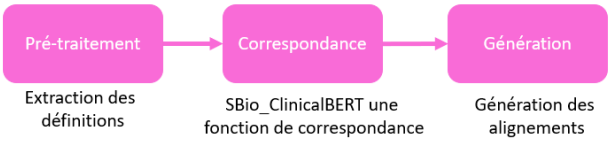

FIGURE 3 – Étapes du processus d'alignement

Bio_ClinicalBERT [1].

3.2 Phases d'alignement

Dans cette section, nous décrivons notre approche pour aligner des éléments provenant de différentes ontologies biomédicales en utilisant notre modèle. Ainsi, ce dernier constite un système central dans le processus de mise en correspondance. Étant donné que les transformeurs fonctionnent comme des modèles de langues, il est nécessaire que les éléments d'ontologie soient définis par des étiquettes (ou des commentaires).
Nous considérons le processus de mise en correspondance comme un problème de similarité où notre modèle (BioS-Transformers) reçoit des éléments extraits des ontologies d'entrée et calcule leur similarité. En fonction du score de sortie, nous concluons s'il existe une correspondance entre les deux éléments.
Pour décrire l'approche du processus d'alignement, ces phases ont été adoptées (voir Figure 3) :
Prétraitement. Les données textuelles ont été extraites des différentes ontologies. Ces données sont liées à tout élément textuel pouvant aider à la comparaison de similarité. Dans notre cas, nous nous sommes limités aux définitions, car les étiquettes ne contiennent pas de données significatives.
Mise en Correspondance. Pour cette étape, nous avons choisi le modèle SBio_ClinicalBERT comme fonction de mise en correspondance. Par rapport à d'autres modèles, ce modèle fournit de bons résultats pour la comparaison d'étiquettes. Cela est dû au fait que ce modèle est déjà entraîné

sur des notes cliniques provenant de MIMIC III.

Génération d'alignement. Les alignements générés sont des correspondances entre un concept de $O1$ et un concept de $O2$.

Cet alignement est maintenu lorsque le score de confiance (score de similarité) est supérieur au seuil de 0,5. D'autres scores vont être étudiés dans des travaux futurs. Si un alignement existe, une nouvelle relation est définie entre le concept $x1$ de $O1$ et le concept $x2$ de $O2$.

Après avoir utilisé notre méthodologie pour aligner les ontologies DOID et DRON [9, 8], nous avons étendu son application à l'alignement d'autres ontologies ou terminologies afin de valider notre approche. Plus spécifiquement, nous avons sélectionné des vocabulaires importants du Metathesaurus UMLS pour cette validation étendue. Nous avons sélectionné tous les vocabulaires en anglais ayant plus de 5000 concepts, ce qui représente un total de 10 vocabulaires. Dans ce travail, nous présentons les résultats pour une paire d'ontologies parmi ces vocabulaires (NCI et PDQ).

4 Expérimentations et Résultats

Nous avons utilisé la version complète de l'UMLS 2023AA. Dans cette partie, nous allons étudier la validité des paires similaires (CUI_1, CUI_2) proposées par nos modèles. Nous proposons deux approches complémentaires :

1. Les deux concepts que nous souhaitons proposer comme équivalents le sont déjà dans l'UMLS (ont le même identifiant CUI dans l'UMLS). Ce qui valide notre approche.
2. Pour les concepts qui n'ont pas le même CUI, nous avons proposé une deuxième approche de validation en recherchant un parent commun et en mesurant la distance entre ce parent et les deux concepts pour savoir la pertinence de cette équivalence.

Par la suite, nous détaillons les étapes de la deuxième approche :

Calcul des Scores de Similarité. Dans cette étape, nous avons sélectionné 1 000 paires de concepts aléatoires (CUI_1, CUI_2) à partir de la paire d'ontologies (NCI, PDQ) afin d'appliquer notre deuxième approche de validation. Ce choix est à cause du très grand nombre de combinaisons possibles entre les concepts des deux ontologies (plus de 820 millions), ce qui rendrait le parcours du graphe pour identifier les parents communs les plus proches extrêmement coûteux en temps de calcul pour chaque paire de concepts. Nous avons ensuite utilisé notre modèle pour calculer les scores de similarité entre ces paires. Nous désignons le score de similarité entre (CUI_1, CUI_2) par S, calculée à l'aide de notre modèle.

Calcul des Distances Entre les Concepts et Tous leurs Parents. L'UMLS a été construit en consolidant de nombreux vocabulaires, mais ceux-ci n'étaient pas entièrement interconnectés depuis les nœuds racines jusqu'aux feuilles. Ils étaient principalement liés à travers les nœuds les plus élevés présents dans les deux vocabulaires, ce qui a posé problème pour la recherche du parent en commun, celui-ci correspondant souvent à la racine ou à des nœuds situés très haut à chaque fois. Pour résoudre ce problème, nous avons opté pour l'utilisation d'un sous-graphe au sein de l'UMLS. Cela nous a permis d'identifier les parents communs au sein de notre sous-ensemble du graphe et sans devoir parcourir l'intégralité du graphe UMLS.

Dans l'UMLS, certains racines des vocabulaires comme MeSH ont été omises, tandis que d'autres étaient incluses dans des cycles avec des concepts plus abstraits tels que "bases de données", etc. Dans notre approche, le calcul des distances que nous avons proposé nécessite l'existence d'une racine. Nous avons opté pour un sous-graphe de l'UMLS composé uniquement des vocabulaires NCI et PDQ. Cependant, le NCI dans l'UMLS, ne possède pas de racine en raison de son inclusion dans un cycle. Bien que le PDQ ait une racine, le graphe combinant les deux vocabulaires n'a pas de racine. Ainsi, nous avons ignoré la relation causant ce cycle et avons retenu la racine du PDQ comme racine commune pour les deux vocabulaires. Cet ajustement nous a permis de calculer la distance entre chaque concept appartenant à l'une de ces ontologies.

Nous commençons à parcourir le sous-graphe à partir d'un concept donné. Supposons que nous commencions avec CUI_1, nous allons traverser le graphe à partir de ce nœud et calculer les distances minimales entre ce nœud et tous ses ancêtres directs et indirects en utilisant l'algorithme 1 que nous proposons. Chaque arête est équivalente à 1. Pour chaque paire de (CUI_1, CUI_2), nous calculons toutes les distances minimales entre CUI_1, CUI_2, et tous leurs ancêtres dans le graphe où $d_{min}(CUI_j, p_i)$ est la distance minimale entre le concept CUI_j et son ancêtre p_i, calculée à l'aide de l'algorithme 1 pour déterminer la distance minimale que nous avons proposée. Nous procédons de même pour CUI_2.

$FindRoots()$ dans l'algorithme 1 est une fonction qui retourne toutes les racines présentes dans le graphe, et $BuildCuiChildrenDict$ est une fonction qui renvoie un dictionnaire contenant tous les enfants directs de chaque nœud du graphe.

Parents Communs de la Paire CUI. À partir des distances minimales entre CUI_1 et tous ses ancêtres et CUI_2 et tous ses ancêtres, nous effectuons l'intersection des ensembles de parents des deux CUIs pour trouver les parents communs entre les deux concepts. Ces parents communs représentent ceux ayant la distance minimale par rapport aux deux concepts (parents plus proches), avec une distance minimale d_1 entre ce parent p et CUI_1 et une distance d_2 entre ce parent p et CUI_2. Une fois l'ensemble des parents communs trouvé, l'étape suivante consiste à évaluer la pertinence de ces relations découvertes. Les distances entre un concept et un parent commun varient en fonction du sous-graphe ou de la hiérarchie en question (la hiérarchie NCI peut aller jusqu'à 24 niveaux de profondeur).

Notre objectif est de trouver la relation entre la distance d'un parent commun à ses fils (CUI_1 et CUI_2) et le score de similarité entre (CUI_1 et CUI_2).

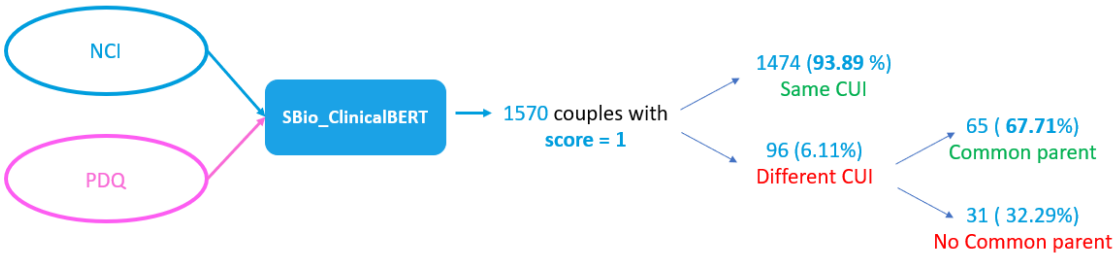

FIGURE 4 – Nombre de couples valides/invalides dans (NCI, PDQ)

Algorithm 1 Find Minimum Distance

$E \leftarrow$ FindRoots($mrrel\_file$)
$root \leftarrow C000$
$d \leftarrow \{\}$
$visited \leftarrow \{\}$
for r in E **do**
    $d[(r, root)] \leftarrow 0$
end for
$cui\_children\_dict \leftarrow$ BuildCuiChildrenDict($mrrel\_file$)
while $|E| > 0$ **do**
    $p \leftarrow E.pop()$
    if p in $visited$ **then**
        continue
    end if
    $visited.add(p)$
    $children\_list \leftarrow cui\_children\_dict.get(p, [])$
    for c in $children\_list$ **do**
        if (p, root) in d **then**
            if (c, root) in d **then**
                $d[(c, root)] \leftarrow \min(d[(c, root)], 1 + d[(p, root)])$
            else
                $d[(c, root)] \leftarrow 1 + d[(p, root)]$
            end if
        end if
        if c not in E **then**
            $E.add(c)$
        end if
    end for
end while
return d

Après avoir calculé la corrélation de Pearson entre les deux variables, la distance $d_{pi}(CUI_1, CUI_2)$ et le score de similarité $S(CUI_1, CUI_2)$, nous avons constaté qu'il existe une forte relation négative entre les deux variables avec une valeur de $r = -0,795$. Cette valeur indique que les concepts qui sont éloignés de leur parent commun dans l'arbre hiérarchique ont une similarité très faible. En revanche, lorsque le parent est proche des deux concepts, ces concepts ont une valeur de similarité élevée. Cela confirme notre hypothèse : plus la distance minimale entre deux concepts et leur parent commun est grande, plus leur similarité est faible.

La figure 5 présente le diagramme de dispersion, ou graphique de corrélation, entre le score de similarité entre les deux concepts $S(CUI_1, CUI_2)$ et la distance moyenne entre ces paires et leurs parents communs. À partir de la figure 5, nous pouvons observer une corrélation négative entre les deux variables. L'écart dans l'intervalle [0,6 ; 0,8] est dû au fait qu'il n'y a pas beaucoup de paires avec un score de similarité dans cette plage dans l'échantillon de données sélectionné au départ.

La figure 4 montre le nombre de couples de CUIs que nous proposons entre (NCI, PDQ) ayant un score de similarité 1 soit 1570 paires. 93% des paires ont déjà le même CUI dans l'UMLS. Cela indique que l'UMLS considère ces CUIs comme équivalents, alors qu'ils représentent en réalité deux concepts distincts issus de deux ontologies différentes. Notre modèle parvient à les distinguer correctement avec un score parfait. Pour les 6% restants, nous avons appliqué notre deuxième approche complémentaire en cherchant un parent commun. Pour 67% de ces paires, nous avons trouvé des parents communs dans le sous-graphe sélectionné, ce qui montre que ces concepts partagent une relation de parenté. En revanche, pour les 33% restants, aucun parent commun n'a été identifié dans ce sous-graphe. Il faudra donc envisager d'autres moyens pour évaluer la validité des alignements proposés. Ces résultats soulignent l'efficacité de notre approche, qui permet de proposer des relations d'équivalence correctes pour plus de 93% des concepts dans un échantillon de 1570 couples (NCI, PDQ), et des stratégies complémentaires pour les valider.

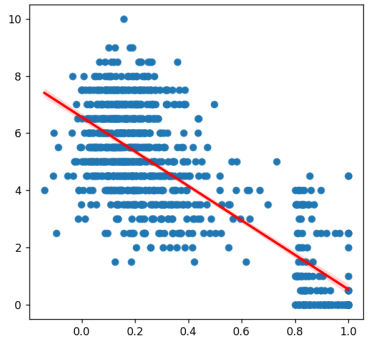

FIGURE 5 – L'évolution des distances moyennes en fonction du score de similarité entre les paires $S(CUI_1, CUI_2)$.

5 Conclusion

Dans cet article, nous abordons le défi de l'alignement d'ontologies sous l'angle de la représentation contextuelle. Notre approche consiste à représenter deux concepts distincts issus de différentes ontologies dans le même espace vectoriel, suivie du calcul de leur similarité sémantique. Cette similarité sémantique permet de créer des correspondances entre des concepts sémantiquement liés. Notre objectif dans cette étude était d'aligner tous les vocabulaires du métathésaurus UMLS afin de mapper sémantiquement différentes entités au sein de l'UMLS, offrant ainsi de nouvelles relations pour répondre à des questions de recherche spécifiques. Nos expériences, menées à travers différents intervalles de scores, démontrent la validité précise des alignements générés. Nous prouvons que les correspondances proposées par nos modèles correspondent authentiquement aux mêmes concepts dans l'UMLS, provenant de vocabulaires différents avec des définitions distinctes, ainsi qu'aux concepts partageant au moins un parent commun, validant ainsi notre approche. Cette méthode s'est avérée efficace pour enrichir l'UMLS.

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] 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.

[3] Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

[4] Xubing Hao, Rashmie Abeysinghe, Jay Shi, and Licong Cui. Exploring pre-trained language models for vocabulary alignment in the umls. In *International Conference on Artificial Intelligence in Medicine*, pages 273–278. Springer, 2024.

[5] 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.

[6] Prodromos Kolyvakis, Alexandros Kalousis, Barry Smith, and Dimitris Kiritsis. Biomedical ontology alignment : an approach based on representation learning. *Journal of biomedical semantics*, 9 :1–20, 2018.

[7] WZ Li, XX Duan, M Wang, XP Zhang, and GL Qi. Multi-view embedding for biomedical ontology matching. 2019 oct 26 presented at : Proceedings of the 14th international workshop on ontology matching collocated with the 18th international semantic web conference ; october 26, 2019. *Auckland, New Zealand*.

[8] Safaa Menad, Wissame Laddada, Saïd Abdeddaïm, and Lina F Soualmia. Biostransformers for biomedical ontologies alignment. In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2023, Volume 2 : KEOD*, pages 73–84. SCITEPRESS, 2023.

[9] Safaa Menad, Wissame Laddada, Saïd Abdeddaïm, and Lina F Soualmia. New siamese neural networks for text classification and ontologies alignment. In *International Conference on Complex Computational Ecosystems*, pages 16–29. Springer, 2023.

[10] Vinh Nguyen and Olivier Bodenreider. Adding an attention layer improves the performance of a neural network architecture for synonymy prediction in the umls metathesaurus. *Studies in health technology and informatics*, 290 :116, 2022.

[11] Vinh Nguyen, Hong Yung Yip, Goonmeet Bajaj, Thilini Wijesiriwardene, Vishesh Javangula, Srinivasan Parthasarathy, Amit Sheth, and Olivier Bodenreider. Context-enriched learning models for aligning biomedical vocabularies at scale in the umls metathesaurus. In *Proceedings of the ACM Web Conference 2022*, pages 1037–1046, 2022.

[12] Vinh Nguyen, Hong Yung Yip, and Olivier Bodenreider. Biomedical vocabulary alignment at scale in the umls metathesaurus. In *Proceedings of the Web Conference 2021*, pages 2672–2683, 2021.

[13] 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.

[14] Maria Taboada, Rosario Lalin, and Diego MartÍnez. An automated approach to mapping external terminologies to the umls. *IEEE Transactions on Biomedical Engineering*, 56(6) :1598–1605, 2009.

[15] Xingsi Xue. A compact firefly algorithm for matching biomedical ontologies. *Knowledge and Information Systems*, 62(7) :2855–2871, 2020.

Session 3 : Conception d'ontologies

Modèle de Données Sémantiques Commun pour l'Espace de Données Européen de l'Energie Omega-X

Fatma-Zohra Hannou[1,*], Lina Nachabe[2,*] , Maxime Lefrançois[2]

[1] EDF R&D, Palaiseau, France
[2] Mines Saint-Etienne, Univ Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, Saint-Étienne, France

fatma-zohra.hannou@edf.fr, lina.nachabe@emse.fr, maxime.lefrancois@emse.fr

Résumé

L'interopérabilité sémantique est un défi important dans les espaces de données de l'énergie, où divers acteurs (fournisseurs d'énergie, gestionnaires de réseau, fournisseurs de services technologiques) échangent des données, tout en ayant l'obligation de se conformer à différents standards et réglementations. Cet article présente le Common Semantic Data Model (CSDM), une ontologie développée dans le cadre du projet européen Omega-X, visant à construire un espace de données fédéré pour le domaine de l'énergie. Le développement de l'ontologie Omega-X a pour objectif d'assurer une interopérabilité sémantique efficace pour tous les acteurs de l'échange de données, et de permettre une exploitation optimale des jeux de données mis à disposition. Le CSDM adopte une approche modulaire et multi-niveau, alignée sur les principes FAIR (Findable, Accessible, Interoperable, Reusable), qui s'appuie sur des ontologies du domaine de l'énergie, ainsi que des standards tels que l'IEC CIM. L'article détaille le processus de développement du CSDM, basé sur les méthodologies AIME et ACIMOV, et présente ses principaux modules. Une évaluation du CSDM est réalisée à travers deux cas d'usage concrets : (1) l'échange de données de flexibilité entre un producteur de données et un fournisseur de services de prédiction et (2) la gestion de la performance des installations photovoltaïques. Ces cas d'usage ainsi qu'une validation technique démontrent la capacité du modèle à assurer l'interopérabilité sémantique et à améliorer la gestion des données dans un espace énergétique fédéré.

Mots-clés

ontologie modulaire, espace de données, énergie, standards, principes FAIR, patron de conception d'ontologies.

Abstract

Semantic interoperability is a major challenge in data spaces, and energy data spaces specifically, where various stakeholders (energy providers, grid operators, and technology service providers) exchange data, while being required to comply with diverse standards and regulations. This paper presents the Common Semantic Data Model (CSDM), an ontology developed as part of the European project Omega-X , which aims to build a federated energy data space. The development of the Omega-X ontology aims to ensure effective semantic interoperability for all data exchange participants and enable the optimal use of available datasets. The CSDM adopts a modular and multi-level approach, aligned with the FAIR principles (Findable, Accessible, Interoperable, Reusable). It builds on existing energy domain ontologies and integrates key standards such as IEC CIM. The paper details the CSDM development process, based on the AIME and ACIMOV methodologies, and presents the main ontology modules that constitute it. The CSDM is evaluated through two concrete use cases : (1) the exchange of flexibility data between a data producer and a prediction service provider, and (2) the performance management of photovoltaic installations. These use cases, along with a technical validation, demonstrate the model's ability to ensure semantic interoperability and improve data management within a federated energy space.

Keywords

modular ontology, data space, energy, standard, FAIR principles, ontology design pattern.

1 Introduction

Dans le contexte de la constante évolution de la transformation digitale, les espaces de données ont émergé comme un outil clé pour permettre un échange de données sécurisé, souverain et interopérable entre plusieurs acteurs, dans différents secteurs. En Europe, les espaces de données sont au cœur de la stratégie Européenne pour les données, qui vise à encourager l'innovation numérique en facilitant le partage de données entre les entreprises, institutions et organisations gouvernementales. Grâce à un écosystème fédéré, ces parties prenantes peuvent participer à l'accélération de l'innovation dans des domaines telles que l'intelligence artificielle, l'industrie 4.0, et la transition énergétique. Un défi majeur pour la concrétisation de ces objectifs

* Ces auteurs ont contribué de manière égale à ce travail.

est la question de l'interopérabilité [19]. L'interopérabilité, et notamment sémantique, est essentielle pour permettre à des acteurs différents de communiquer et d'exploiter correctement les données partagées. Ceci est particulièrement complexe dans le secteur de l'énergie, où de multiples acteurs (fournisseurs d'énergie, gestionnaires de réseau, organismes de régulation, et fournisseurs de technologies) doivent collaborer tout en se conformant à des régulations et standards nationaux et internationaux. L'interopérabilité au sein d'un espace de données requiert un cadre commun de représentation de la donnée qui peut être assuré par l'emploi d'ontologies. Cependant, la conception d'une ontologie pour un espace de données présente de nombreux défis. D'abord, un contexte d'échange de données dynamique implique plusieurs évolutions, des interactions à travers des structures complexes, et une intégration de sources de données hétérogènes. Ensuite, chaque secteur est régi par diverses normes de données, qui ne sont souvent pas des ontologies. Par exemple pour le domaine de l'énergie, l'IEC CIM (Common Information Model) [21]. Concevoir une ontologie pour l'espace de données qui conserve la compatibilité avec ces normes est un défi. La complexité de ces facteurs nécessite une conception d'ontologie robuste et adaptable.

A partir de ces observations, nous définissons les exigences (**E**) suivantes pour le développement d'une ontologie permettant un partage efficace et évolutif des données dans un espace de données de l'énergie.

- **E1 Modularité** : L'ontologie doit être structurée de manière à permettre l'évolution indépendante de différents composants tout en maintenant une cohérence à l'échelle de l'espace de données.
- **E2 : Représentation multi-niveaux :** l'ontologie doit permettre différents niveaux d'abstraction, allant des concepts généraux du domaine de l'énergie aux applications spécifiques à un cas d'usage donné.
- **E3 : Conformité aux normes et standards** : L'ontologie doit s'aligner avec les standards existants tout en offrant la possibilité d'extensions pour s'adapter à des évolutions et spécificités des cas d'usage.
- **E4 : Support à la création de graphes de connaissances** : L'ontologie doit être conçue pour optimiser la construction de graphes de connaissances, facilitant ainsi l'intégration des données, le requêtage, le raisonnement et l'analyse.

Le projet Européen Omega-X vise à développer un espace de données de l'énergie en se focalisant sur quatre grandes familles de cas d'usages : la flexibilité, l'énergie renouvelable, les communautés locales d'énergie ainsi que l'électromobilité. Dans ce cadre, et pour répondre aux exigences définies pour une ontologie du domaine de l'énergie, nous développons l'ontologie Omega-X comme un modèle de données sémantique commun (Common Semantic Data Model-CSDM) visant à renforcer l'interopérabilité sémantique au sein de l'espace de données. La suite de cet article est organisée en 4 sections. La section 2 présente les travaux existants utilisés comme ressources lors du développement du CSDM. Ceci inclut à la fois les standards du domaine de l'énergie mais également les ontologies existantes. La section 3 détaille les principaux modules constituant le CSDM. La section 4 présente les travaux de validation techniques ainsi que deux cas d'usages qui évaluent le CSDM dans les domaines de la flexibilité et de l'énergie renouvelable. Enfin, la section 5 conclut et discute la contribution.

2 Travaux antérieurs

Plusieurs normes internationales ont été élaborées pour faciliter l'interopérabilité dans les réseaux électriques intelligents et les systèmes énergétiques. La norme ***IEC 61850*** [3] définit des protocoles de communication permettant l'échange de données entre différents équipements situés dans une sous-station, tels que les dispositifs de protection, de contrôle et de mesure, ainsi que les dispositifs électroniques intelligents. La norme générale IEC 61970 [21] (***CIM - Common Information Model***) fournit un modèle de données et un cadre d'échange d'informations pour la gestion des systèmes électriques, allant de la production à la distribution en passant par les opérations de marché. Le profil EUMED (*European My Energy Data*) est un profil développé dans le cadre des normes IEC CIM, particulièrement en relation avec l'échange de données énergétiques pour le secteur européen de l'énergie. Ce profil se repose sur des standards clés, dont IEC 62325-451-10 et facilite l'accès à des données telles que la consommation d'énergie [6]. Citons aussi la norme **DLMS/COSEM** (*Device Language Message Specification/Companion Specification for Energy Metering*), définie dans la norme internationale IEC 62056 qui permet de modéliser les données des compteurs, afin d'assurer l'interopérabilité dans les systèmes de gestion [11].

Les normes sont indispensables pour garantir l'interopérabilité technique au sein des réseaux intelligents. Cependant, elles ne suffisent pas pour garantir une communication efficace dans les systèmes complexes des réseaux intelligents. Il ne s'agit pas seulement d'assurer que les systèmes puissent échanger des données de manière techniquement compatible, mais aussi que ces données soient compréhensibles et exploitables de manière cohérente par tous les acteurs impliqués. C'est là qu'intervient l'interopérabilité sémantique qui peut être assurée en développant des ontologies modélisant le domaine visé. Dans la suite de cette section nous nous concentrons sur celles basées sur l'ontologie standard ETSI SAREF [12] ainsi que sur celles développées dans le cadre des projets européens liés à l'énergie.

L'ontologie **SARGON** est développée dans le cadre du projet N5GEH, et vise à améliorer la compréhension sémantique des équipements intelligents dans les bâtiments et les Smart Grids. Elle s'appuie sur des classes de l'ontologie SAREF et intègre des éléments des modèles de données IEC CIM et IEC 61850 [13]. Bien que l'ontologie SARGON couvre les équipements et les infrastructures des systèmes énergétiques intelligents, elle ne modélise pas l'échange de données entre ces systèmes ni les profils de flexibilité et de demande/réponse énergétiques.

L'ontologie **INTERCONNECT** développée dans le cadre du programme H2020, propose 9 modules ontologiques qui visent à connecter de manière interopérable les maisons intelligentes, les bâtiments et les réseaux. Ces modules permettent de décrire les séries temporelles, la flexibilité de la demande et le standard S2 pour modéliser les profils de flexibilité. Ils reposent sur l'ontologie SAREF et couvrent des domaines comme la gestion de consommation, la topologie des réseaux et les appareils [9]. Ces travaux ont été intégrés à une nouvelle version de l'extension **SAREF4ENER** de SAREF. Cette extension repose sur le IEC CIM et permet aux clients de gérer des appareils domotiques via un gestionnaire d'énergie. Elle prend également en charge des fonctionnalités avancées comme la surveillance de la consommation en temps réel et la gestion de la flexibilité de la consommation énergétique [2]. De plus, **SEAS** (Smart Energy Aware Systems) [17] est développée dans le cadre du projet ITEA 12004 SEAS, et se concentre sur la définition des systèmes électriques qui produisent, consomment ou stockent de l'électricité. Elle inclut des modules pour décrire les objets d'intérêt, les évaluations, et les systèmes interconnectés. SEAS propose également des ontologies spécifiques pour des domaines comme les batteries, les panneaux photovoltaïques, les véhicules électriques, ainsi que pour les profils de flexibilité, la prévision et la comptabilité des prix de l'énergie. Cela permet d'améliorer la description des systèmes énergétiques à travers des modules adaptés aux différents composants du Smart Grid. En effet, SEAS ne modélise pas de manière explicite les mécanismes d'échange de données entre les systèmes énergétiques interconnectés, tels que les flux d'information relatifs à la consommation, la production ou encore les prévisions de charge. Dans le but de numériser le secteur de l'énergie afin d'optimiser son efficacité opérationnelle, le projet PLATOON financé par l'UE dans le cadre du programme Horizon 2020 propose l'ontologie **SEDMOON**, réutilisant SAREF, SEAS, et OntoWind, pour les besoins spécifiques des pilotes et des cas d'utilisation dans le domaine de l'énergie. Elle est constituée de 18 modules, organisés en 5 groupes principaux, couvrant des aspects tels que les bâtiments intelligents, la gestion de l'énergie dans les smart grids, la génération d'énergie renouvelable, et les équipements HVAC [1]. Une extension de cette ontologie a été proposée dans le projet **ENERSHARE** et comprend 18 modules conçus pour faciliter l'interopérabilité sémantique dans le domaine de l'énergie. Elle inclut le module Digital Twin, le module Energy Resource qui fournit une taxonomie des types de ressources énergétiques, le module Event qui étend la classe des événements de PLATOON pour intégrer des événements planifiés et des événements électriques du réseau. De plus, le module Market fournit une taxonomie pour les types de marchés, le module Player étend l'ontologie SEAS Player pour inclure des rôles, le module Property qui étend les propriétés SEAS, et le module System qui englobe les systèmes et les équipements en réutilisant les concepts déjà existants dans SEAS.

Les ontologies énergétiques présentent une complémentarité en termes de couverture de domaine. Par contre, les ontologies actuelles n'incluent pas une modélisation exhaustive des mécanismes d'échange de données, notamment pour des flux d'informations dynamiques tels que les prévisions de charge, les profils de flexibilité en temps réel et les transactions énergétiques complexes.

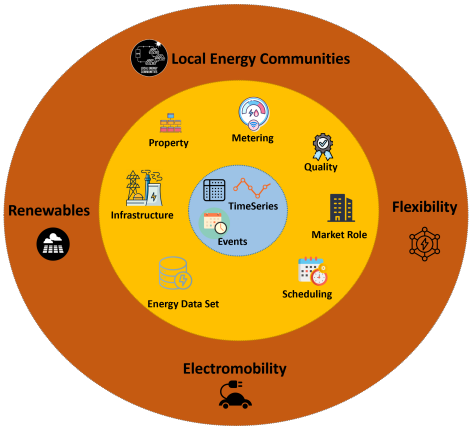

FIGURE 1 – Aperçu de l'ontologie Omega-X, constituée de 12 modules, structurés en 3 niveaux d'abstraction : Haut niveau (en bleu), Domaine de l'énergie (en jaune), et applications dans le domaine de l'énergie (en orange)

L'approche recommandée par W3C consiste à réutiliser au maximum les concepts en étendant les classes existantes au lieu de les modifier, en ajoutant des sous-classes et en associant de nouvelles propriétés. Cette méthode permet de préserver l'intégrité des modèles tout en offrant des adaptations spécifiques.

3 L'ontologie Omega-X

3.1 Méthode de développement

L'ontologie OMEGA-X a été développée en appliquant la méthodologie AIME (*Agile Interaction Model-based ontology development Methodology*) [18], mise au point dans le projet pour permettre une approche agile et modulaire de création d'ontologies adaptées aux espaces de données. La méthodologie AIME repose sur plusieurs principes : la considération des standards de référence et d'ontologies existantes (**E3**), la définition des cas d'usage et des modèles d'interaction associés (**E2**), la conception de la modularité d'ontologie (**E1**), et l'intégration continue à travers des outils d'automatisation. L'ontologie OMEGA-X (version actuelle 1.1) [1] illustrée dans la figure 1 comporte 12 modules organisés dans une architecture modulaire et multi-niveaux permettant de structurer les connaissances du domaine énergétique selon plusieurs niveaux d'abstraction. Dans son niveau supérieur (noyau bleu dans la figure 1), elle permet de capturer la structure des ensembles de données échangés dans l'espace de données. Ensuite, au deuxième niveau d'abstraction (en jaune dans la figure 1), 7 modules transversaux représentent des connaissances communes à plusieurs domaines dans le secteur de l'énergie. Cela com-

1. https://w3id.org/omega-x/

prend entre autres l'infrastructure des sites de production, du réseau, les propriétés du domaine, les métriques de qualité normalisées, ou encore les caractéristiques des ensembles de données de l'énergie. Au dernier niveau (orange dans la figure 1), 4 modules permettent de représenter les spécificités des quatre familles de cas d'usage de l'espace de données Omega-X : énergies renouvelables, flexibilité, communautés locales d'énergie, et électromobilité.

Pour se conformer au principe (**E4**) de faciliter la construction de graphes de connaissances, le pattern "Kind of X and X of Interest" [15] a été appliqué. Cela permet de séparer les considérations entre les développeurs de l'ontologie, qui fournissent des taxonomies de propriétés génériques et de systèmes du domaine de l'énergie (usage de *kindOfXX*) , et les créateurs de graphes de connaissances qui matérialisent des instances de propriétés et systèmes relatives à leurs cas d'usage (usage de *XOfInterest*).

Le site web `https://w3id.org/omega-x/` héberge le CSDM ainsi que sa documentation. Tous les URIs des modules sources (`https://w3id.org/omega-x/repository`) sont permanentes via *w3id*.

3.2 Événements et Séries Temporelles

Le module Événements et Séries Temporelles (**Events and Time Series, ETS**) [2] constitue le module de niveau supérieur du CSDM (diagramme CHOWLK [5] illustré dans la figure 2). C'est le module principal permettant de capturer la structure des jeux de données et de leur associer les métadonnées pertinentes. Dans ETS, le concept principal est ets:ValueSet ou ensemble de données, spécifié à travers quatre sous-types d'ensembles de données définis :

- **Le point de données** : ets:DataPoint décrit une données élémentaire.
- **L'événement** : ets:Event décrit une occurrence d'un changement dans l'environnement.
- **Une série temporelle** : ets:TimeSeries est un conteneur pour une série de ets:ValueSet horodatées. Les éléments constituant une série temporelle ont obligatoirement une empreinte temporelle. Si ces éléments sont espacés d'un intervalle régulier, ets:step permet de le renseigner.
- **Une collection de données** : ets:DataCollection est un conteneur de ets:ValueSet, ordonnées ou non. Si les éléments sont ordonnés, chacun aura un ets:rank spécifié.

Les valeurs de points de données et d'événements sont indiquées à travers la classe ets:DataValue, qui permet de renseigner une valeur soit quantitative via ets:value, soit qualitative via un individu issu de la taxonomie des catégories de valeurs qualitatives (détaillée dans la section 3.4). Les collections de données et les séries temporelles peuvent contenir d'autres ensembles de données, qu'ils soient élémentaires (ets:DataPoint et ets:Event) ou complexes.

Gestion de la temporalité Un point de données peut être associé à un ets:dateTime qui capture le temps réel auquel la valeur est applicable. De manière analogue, les événements sont décrits par un ets:triggeringTime qui reflète le temps de déclenchement réel. Une deuxième information temporelle peut être renseignée pour préciser le ets:creationTime, qui est le temps correspondant à la création digitale d'un ensemble de données. Le contexte temporel permet quant à lui de décrire une durée de validité temporelle de l'ensemble de données.

Gestion des propriétés Les ensembles de données comportent des valeurs qui peuvent être associées à des propriétés (voir section 3.4). Un événement ou un point de données est associé à une propriété prop:PropertyOfInterest unique. Une collection/série peut être liée à une propriété par la relation ets:allAboutProperty, auquel cas tous ses membres hériteront de cette relation. Alternativement, une ou plusieurs relations ets:someAboutProperty peuvent être utilisées, ces relations se propageant dans l'autre direction (du contenu vers le contenant).

Métadonnées sur la qualité Un ensemble de données décrit la qualité d'un ensemble de données à travers le module **Qualité** (section 3.5). Ce module permet d'exprimer divers attributs de qualité, tels que la précision, la complétude, la cohérence ou encore la provenance des données.

3.3 Ensembles de données énergétiques

Le module Ensemble de données énergétiques **Energy Data Sets** [3] illustré sur la figure 3, associe un ensemble de valeurs à un échange dans l'espace de données Omega-X, constituant ainsi un eds:EnergyDataSet. Dans la mesure où le partage des ensembles de données dans des environnements complexes — notamment dans le secteur de l'énergie — nécessite une description précise du contexte d'échange, eds:EnergyDataSet peut être lié à un contexte d'échange. Ce dernier décrit les acteurs du marché impliqués dans l'envoi et la réception des données et s'accompagne d'un contexte technique qui spécifie, par exemple, le format des données, les exigences techniques ou d'autres caractéristiques pertinentes. De plus, un point d'évaluation, désigné sous le terme eds:EvaluationPoint, est associé à cet ensemble de données. Cette notion s'inspire du profil EU-MED Metering, où le point d'utilisation (Usage Point) est défini comme un « point logique ou physique dans le réseau auquel des relevés ou des événements peuvent être attribués » , permettant ainsi de regrouper les séries temporelles selon leur origine [4]. Toutefois, pour offrir une plus grande flexibilité et précision dans l'attribution des données, le point d'évaluation peut être étendu à des entités plus larges, tels qu'un système global de gestion de l'énergie d'un quartier ou une plateforme centralisée.

3.4 Propriété

Le module propriété, (**Property, PROP**) [4] permet de définir les caractéristiques du monde réel décrites par des

2. `https://w3id.org/omega-x/ontology/EventsTimeSeries`

3. `https://w3id.org/omega-x/ontology/EnergyDataSet`

4. `https://w3id.org/omega-x/ontology/Property`

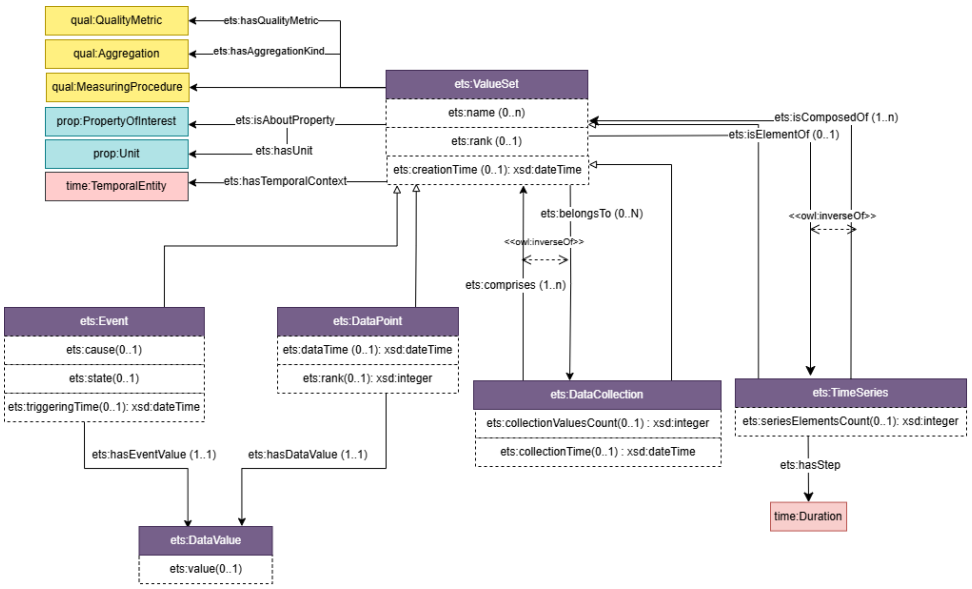

FIGURE 2 – Diagramme Chowlk du module **Evénements et Séries Temporelles**

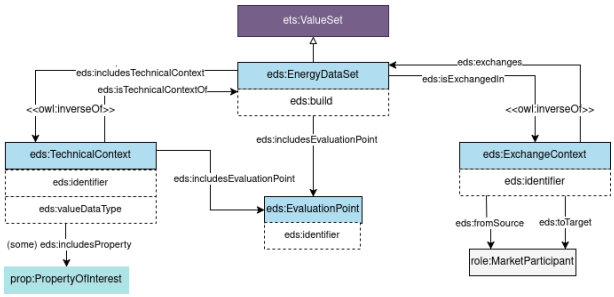

FIGURE 3 – Diagramme CHOWLK du module **ensemble de données énergétiques**

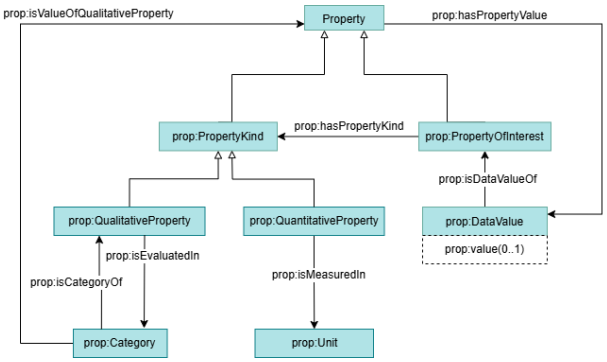

FIGURE 4 – Diagramme Chowlk du module **Propriété**

valeurs dynamiques dans les ensembles de données (module **ETS**), ainsi que les caractéristiques à valeurs statiques (valeur relativement fixes dans le temps, renseignées via prop:DataValue). En appliquant le patron de conception "XKindOfX XOfInterest", **Prop** distingue deux sous-classes de propriétés :

— **prop:PropertyKind**, ou **propriétéType** : Superclasse permettant de définir une taxonomie d'archétypes de propriétés utilisées dans le domaine de l'énergie. Tous les individus de cette classe sont des *skos :concept*. Les relations *skos :narrower*, et *skos :broader* permettent de définir des propriétés plus spécifiques/génériques.

— **prop:PropertyOfInterest**, ou **propriétés d'intérêt** : sont les propriétés décrivant des caractéristiques d'entités du monde réel (environnement, systèmes dans dans le module **infra** décrit en Section 3.7).

les propriétés peuvent être quantitatives (prop:QuantitativeProperty) et avoir donc des valeurs quantitatives (prop:value), et potentiellement des unités de mesure associées (prop:unit). Elles peuvent également être qualitatives (prop:QualitativeProperty), dont les valeurs se définissent via une catégorie de valeurs (individus). Les individus de prop:Category sont des valeurs possibles de propriétés (prop:Category est sous classe de prop:DataValue).

3.5 Qualité

Le module Qualité **Quality, Qual**[5] permet de fournir des métadonnées aux ensembles de données en caractérisant leur qualité. La classe principale QualityAssessment est spécifiée par 3 sous-classes :

— **qual:QualityMetric** : Indicateurs quantitatifs permettant de mesurer la qualité des données (ex : taux d'erreur, précision, latence). Ces métriques évaluent des attributs de qualité (qual:QualityAttribute) répertoriés par la norme *ISO 25012* [14] telles que la précision, la complétude, l'exactitude, ou la cohérence.

5. https://w3id.org/omega-x/ontology/Quality

— **qual:MeasuringProcedure** : Procédure utilisée pour obtenir l'ensemble de données ets:ValueSet (ex : Prédiction, Simulation, Observation).

— **qual:Aggregation** : Mécanismes de calcul pour regrouper les métriques et produire des valeurs synthétiques (ex : Moyenne, Maximum, Médiane, Cumulatif). Contextes d'agrégation : permet de spécifier les dimensions de l'aggrégation temporelle (ex : sur une journée par exemple), spatiale (ex : sur une région) ou logique (ex : sur un ensemble d'équipement).

Ce module énumère plusieurs dimensions de qualité qui permettent aux utilisateurs des ensembles de données de qualifier/quantifier la fiabilité de ces derniers pour une exploitation efficace. Le choix des services d'analyses de données et d'application IA dépend de ces métriques.

3.6 Rôles Energétiques

Le module role **EnergyRole, ERO** [6] est dédié à la définition des parties prenantes, de leurs comportements attendus et des relations qu'elles entretiennent avec d'autres éléments impliqués dans les processus commerciaux du marché de l'énergie. Ce module définit la classe des (participant), mais elle ajoute les rôles commerciaux (rôle) et leurs relations (association). Les rôles sont ensuite détaillés à travers des classifications spécifiques aux domaines et aux cas d'usage. Un *Participant au marché* est défini comme l'identification d'une partie impliquée dans les processus commerciaux du marché de l'énergie. Un *Rôle sur le marché* désigne l'identification du comportement prévu d'un participant au marché dans un processus commercial [7].

3.7 Infrastructure

Dans le domaine de l'énergie, une topologie d'infrastructure **Infrastructure, Infra** [7] simplifiée est essentielle pour permettre la fourniture de services intelligents, en offrant une description flexible des composants du système et de leurs interrelations. Ainsi, le module illustré dans la figure 6 permet de modéliser l'architecture d'un système énergétique, en définissant notamment des systèmes physiques et virtuels, des équipements et des sites. Ce module réutilise la connexion entre différents systèmes décrite dans l'extension SAREF4SYST de SAREF, en l'étendant et en précisant le sens de la connexion à l'aide des relations infra:isConnectedTo et infra:isConnectedFrom. Pour distinguer les systèmes de leurs types, le modèle "Kind of X/X of Interest" est utilisé [15]. Ainsi, un système peut être un système d'intérêt, représentant un système spécifique au sein de l'infrastructure réelle ou un type de système, qui fournit une taxonomie pour classer les types d'équipements et les sites d'infrastructure. Un système d'intérêt peut être localisé et associé à un acteur du marché de l'énergie. De plus, un système peut être décrit par des propriétés souvent issues de sa fiche technique ou de sa configuration initiale. Enfin, des modules spécifiques peuvent être créés pour détailler l'infrastructure de familles de cas d'usage particulières, tout en restant alignés avec l'ontologie générale.

3.8 Planification

Le module plannification **Scheduling, SCHED** [8] est inspiré du module planning dans SEDMOON mais généralisé pour différents cas d'usage. La classe cœur est le calendrier qui est décrit par un plan pour réaliser une tâche spécifique. Dans le domaine énergétique, un calendrier peut être utilisé pour gérer la consommation et la production d'énergie, optimiser l'efficacité et définir des ensembles de valeurs dans les systèmes ou services.

3.9 Les modules spécifiques aux sous-domaines

Plusieurs modules d'application sont définis afin de spécifier les infrastructures ou les opérations particulières associées à chaque cas d'usage.

Le module Énergie Renouvelable Solaire SOLAIRE Ce module décrit la topologie des centrales photovoltaïques ainsi que la configuration de leurs équipements. Il réutilise des concepts déjà définis dans SEAS, tels que les panneaux solaires et les modules solaires. De plus, il spécifie les relations entre les équipements, notamment entre la sous-station, la station onduleur et la boîte de combinaison, ainsi que les caractéristiques de ces équipements, comme la tension d'entrée DC maximale. Enfin, il prend en compte les connexions entre les équipements, telles que définies dans le label IEC 61850.

Le module Énergie Locale LEC décrit les équipements et services des communautés énergétiques locales (LEC). Il intègre des dispositifs tels que les compteurs, batteries et transformateurs, en réutilisant leurs propriétés issues des fiches techniques. Il couvre également divers services, notamment le calcul des pertes qu'elles soient en eau, thermiques ou électriques , la gamification, ainsi que d'autres fonctionnalités avancées et d'autres fonctionnalités. De plus, il modélise les équipements connectés au réseau, tels que les bus et les transformateurs, assurant une représentation cohérente de l'infrastructure énergétique.

Le module Flexibilité FLEX Le module flexibilité étend les modules transversaux sur 3 dimensions :

— L'infrastructure : spécifie les systèmes du réseau électrique (station primaire, secondaire,..) définis par la norme *IEC 62746* [8].

— Les données : les ensembles de données échangés sont qualifiées en 3 catégories : la catégorie des offres de flexibilité, des demandes et des ordres de flexibilité.

— Les propriétés des cas d'usage de la flexibilités, leurs unités, et catégories qualitatives associées.

Electromobilité EM décrit l'infrastructure des stations de recharge des véhicules électriques.

6. https://w3id.org/omega-x/ontology/EnergyRole
7. https://w3id.org/omega-x/ontology/Infrastructure
8. https://w3id.org/omega-x/ontology/Scheduling

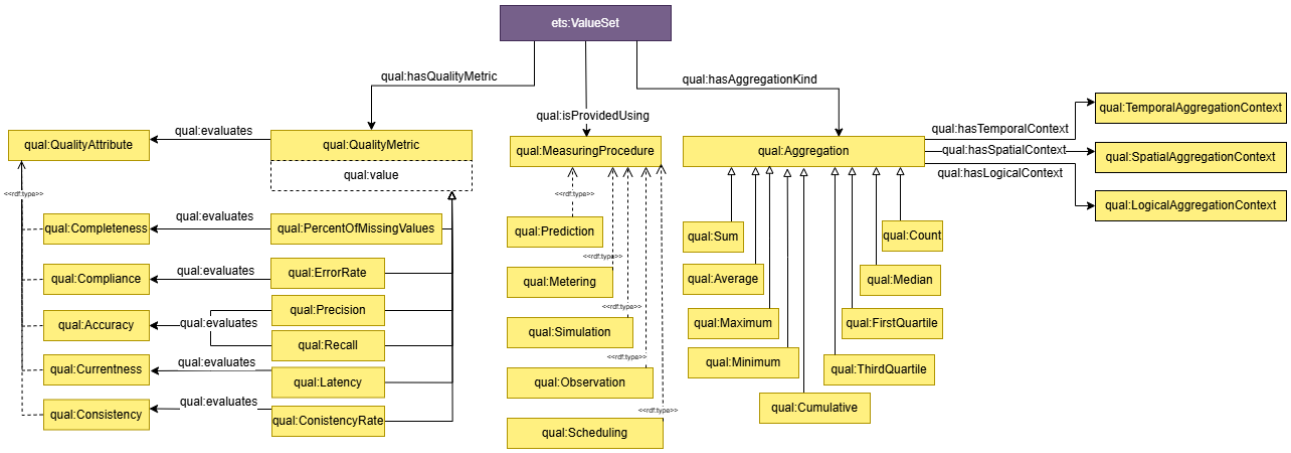

FIGURE 5 – Diagramme Chowlk du module **Qualité**

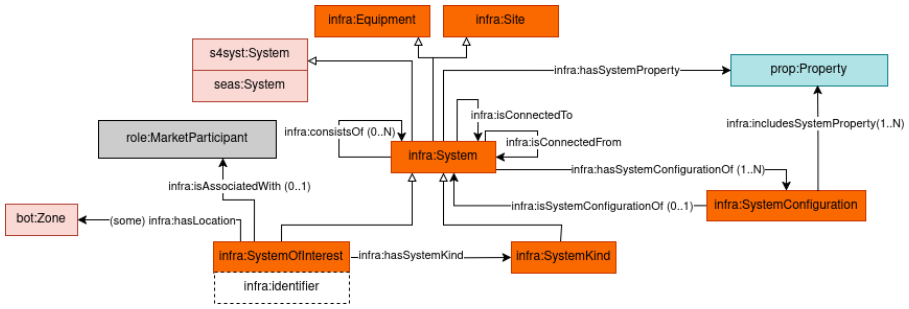

FIGURE 6 – Diagramme CHOWLK du module **Infrastructure**

4 Validation & cas d'usage

4.1 La validation de l'ontologie

La validation structurelle de l'ontologie repose sur son architecture modulaire, une stratégie de versionnage et une documentation détaillée. L'ontologie est structurée en plusieurs modules : un module principal garantissant l'interopérabilité entre espaces de données, des modules spécifiques au domaine de l'énergie ainsi que des modules orientés cas d'usage. Le versionnage suit les principes d'OBO Foundry [22], avec des identifiants uniques (*IRI*) et un suivi précis des mises à jour via des métadonnées. La documentation accompagne chaque module avec des diagrammes de classes, un glossaire, des recommandations et une liste de questions de compétence, facilitant sa réutilisation et assurant son alignement avec les standards. En complément, la validation syntaxique est réalisée via un script Python assurant la conformité aux spécifications Turtle, tandis que la validation sémantique repose sur des requêtes SPARQL vérifiant que l'ontologie réponde aux questions de compétences définies par les experts métiers.

De plus, pour garantir la conformité aux principes FAIR et à l'outil FOOPS !, plusieurs éléments de métadonnées ont été définis. Des URI persistants sont utilisés via w3id.org, chaque version d'ontologie ayant un identifiant unique (owl:versionInfo) et des métadonnées normalisées (description, creator). La disponibilité (Accessible) est assurée par une négociation de contenu RDF et HTML et une publication via des protocoles ouverts (HTTP/HTTPS). Pour garantir l'interopérabilité, l'ontologie est disponible en format Turtle, et des vocabulaires existants comme Dublin Core sont réutilisés pour déclarer les métadonnées. Enfin, pour assurer la réutilisabilité, une documentation HTML est fournie avec des métadonnées détaillées (rdfs:label, rdfs:comment, rdfs:seeAlso), une licence explicite (MIT) et des informations sur la provenance des données (creator, :contributor). Les principaux éléments à améliorer sont l'enregistrement de l'ontologie et de son préfixe dans un registre public, ainsi que l'ajout du nom du publieur.

4.2 L'évaluation de l'ontologie par cas d'usage

Le modèle sémantique commun d'Omega-X a été utilisé pour permettre l'intéropérabilité sémantique au sein du projet, et a été appliqué à ce titre dans 9 sites pilotes. Un processus de sémantisation a permis la création de graphes de connaissances pour différents cas d'usage, dans différents sites pilotes du projet Omega-X. Les deux sections suivantes présentent un aperçu des implémentations pour des cas d'usage de flexibilité et d'énergie renouvelable.

4.2.1 Flexibility UC

Dans le domaine de l'énergie, la flexibilité est définie par la capacité d'un système électrique à ajuster sa consommation ou sa production à une demande en électricité variable, anticipée ou non prévue. Dans le projet Omega-X, le site pilote situé dans la ville de MAIA au Portugal a pour objectif d'offrir la flexibilité aux utilisateurs du réseau électrique, tout en optimisant des fonctions de coût. A travers l'espace de données, plusieurs données sont échangées couvrant la production et la consommation électrique des immeubles ou mesurée sur le réseau électrique (station secondaire), profils énergétique, résultats de services de prédiction, etc. En plus de ces données dynamiques, un ensemble de fiches descriptives est également disponible, notamment pour décrire l'architecture réseau, les équipements des différentes localisations, ou encore les caractéristiques des équipements électriques. L'ensemble de ces données est hébergé dans un dépôt Github, et des mises à jour régulières permettent de fournir les mesures récentes.

Dans ce cadre, un travail a été effectué avec le fournisseur des données afin de créer un pipeline de sémantisation (script python) automatiquement déclenché à chaque mise à jour pour enrichir le graphe de connaissance. Le graphe de connaissances initial contient toute l'architecture réseau du site pilote, ainsi que les équipements disponibles et leurs caractéristiques, et est progressivement incrémenté par les séries temporelles enregistrées. Chaque série temporelle ets :TimeSeries contient toutes les collections de données (une ets :DataCollection par pas de temps), et chaque collection de données est constituée des mesures associées à chaque équipement. Une mesure (ets :DataPoint porte sur une propriété ets :PropertyOfInterest différente). La figure 7 illustre un diagramme CHOWLK d'un graphe échangé pour évaluer le service de prédiction de flexibilité à la tour *Torre Lidador* situé à Maia. Plusieurs équipements de l'immeuble, tels que des HVAC sont sujets à des planifications pour optimiser la flexibilité.

La Figure 8 illustre un extrait du graphe de connaissance créé à partir des données de consommation électriques.

4.2.2 Cas d'usage des maintenances des panneaux photovoltaïques

Pour assurer la maintenance des grandes installations photovoltaïques, il est essentiel de disposer d'un système de supervision performant, capable de diagnostiquer l'état de santé des différents composants, de détecter et signaler toute sous-performance significative, et de fournir des recommandations sur les actions de maintenance à entreprendre. Pour cela, deux types de données sont nécessaires : les données statiques, qui décrivent l'infrastructure, notamment la disposition des panneaux solaires, les caractéristiques des onduleurs et les connexions électriques ; les données dynamiques, collectées périodiquement, qui incluent des mesures clés tels que le courant, la tension, l'irradiance et d'autres paramètres de performance.

Dans le cadre du projet OMEGA-X, EDF REN est un fournisseur de données de production d'énergie solaire. Ces données sont labellisées selon le protocole IEC 61850, qui définit la topologie des équipements connectés, la propriété mesurée, l'unité de mesure et le format des valeurs. La figure 9 illustre un extrait d'un exemple de graphe de connaissance, mettant en évidence l'infrastructure de ce parc solaire. Ce graphe réutilise le module de domaine Infrastructure 3.7 pour spécifier qu'il s'agit d'un système d'intérêt, ce qui permet ensuite de sélectionner tous les systèmes d'intérêt (SystemOfInterest). Ensuite, il réemploie le module du cas d'usage des énergies renouvelables solaires (solaire) afin de préciser les types de systèmes, tels que l'onduleur (inverter), le site,le poste de transformation (substation), et la station d'onduleurs (inverter station), ainsi que leurs connexions. Par exemple, un onduleur (inverter) est défini comme un sous-système d'une station d'onduleurs (solar :subSystemOfInverterStation). De plus, le graphe décrit les propriétés techniques des équipements en s'appuyant sur les fiches techniques (datasheets). Cet extrait indique que l'onduleur possède une puissance d'entrée maximale en courant continu de 268250 W.

4.2.3 Statistiques d'application

Le CSDM a été évalué dans le projet OMEGA-X à travers son application dans 9 sites pilotes où les données brutes ont été sémantisées, pour se conformer aux différents modules présentés dans la section 3. Parmi ces sites, 5 sites ont sémantisé la totalité de leurs jeux de données, 3 sites ont sémantisé un jeu de données, et 1 site n'a pas encore mis en œuvre la sémantisation par manque de temps. Différentes stratégies de sémantisation ont été utilisées : des stratégies qui reposent sur les règles RML [10] et d'autres sur RDFlib [20], ainsi qu'un site pilote qui s'est appuyé sur SPARQL-Generate [16]. L'usage du CSDM dans ces différents sites a démontré son efficacité pour l'interopérabilité sémantique. Pour les fournisseurs de services, l'usage d'un même modèle de données a réduit les coûts de préparation des données hétérogènes à intégrer dans leurs services. D'autre part, l'application d'un modèle unifié, riche en métadonnées, a permis aux producteurs de données d'augmenter la visibilité et la réutilisation de leurs jeux de données.

5 Conclusion

Le Common Semantic Data Model (CSDM) développé dans le cadre du projet Omega-X répond aux exigences d'interopérabilité au sein de l'espace de données de l'énergie, grâce à une ontologie modulaire et multi-niveau alignée sur les standards du domaine et les principes FAIR. Dans cet article, les principaux modules du CSDM ont été présentés, sur 3 niveaux d'abstraction, assurant l'interopérabilité à différentes échelles. D'abord, au sein des familles de cas d'usage du domaine de l'énergie (électromobilité, flexibilité, énergie renouvelable, communautés locales d'énergie). Ensuite, des modules transverses assurent l'interopérabilité dans des échanges de données du domaine de l'énergie, en offrant notamment deux taxonomies (propriétés et systèmes). Enfin, un module central permet de décrire des ensembles de données, leurs structures, leurs propriétés et leurs métadonnées de qualité, assurant ainsi un niveau d'interopérabilité et d'intégration avec d'autres

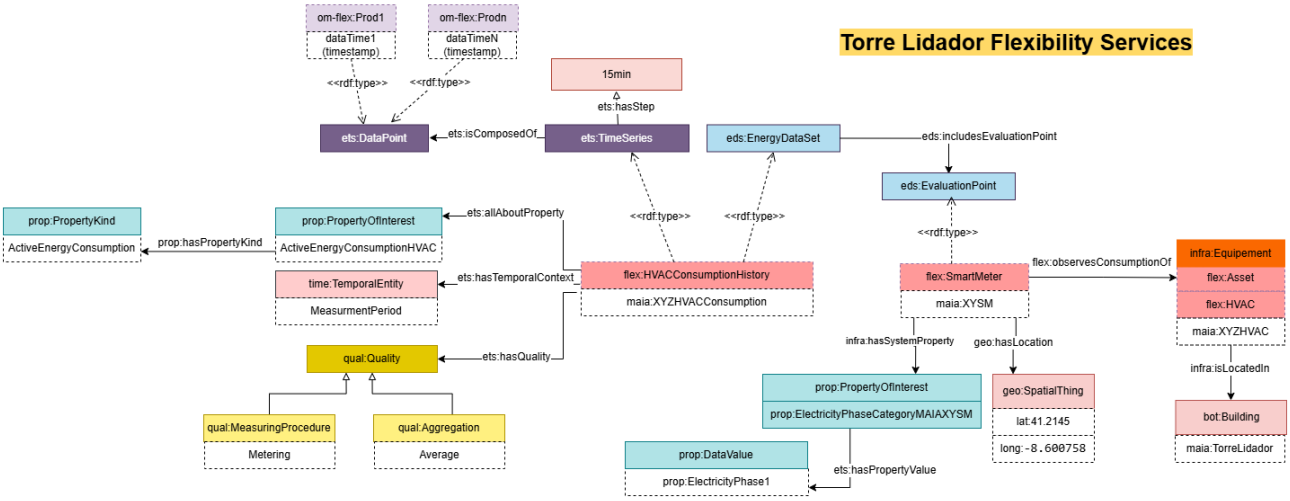

FIGURE 7 – Aperçu du diagramme CHOWLK d'un graphe de connaissance du site pilote de MAIA sur les cas d'usage de flexbilité. Le service de prédiction de flexibilité est appliqué à la tour *Torre Lidador* à MAIA.

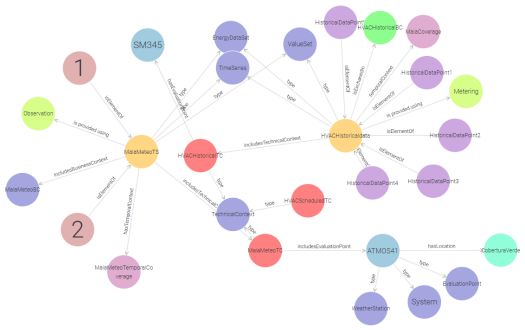

FIGURE 8 – Aperçu du graphe de connaissance de MAIA, pour le cas d'usage de la flexibilité électrique

ensembles de données, au-delà du secteur. Le développement du CSDM s'est accompagné de plusieurs défis notamment ceux relatifs à la grande hétérogénéité des schémas des fournisseurs de données et technologies, nécessitant un travail d'harmonisation et d'alignement avec les standards adoptés. L'applicabilité du CSDM pour l'interopérabilité sémantique a été démontrée grâce à des implémentations dans 8 sites pilotes du projet, dont 2 cas d'usages présentés dans cet article. Les travaux de sémantisation ont nécessité des efforts d'adaptation aux architectures de production des données, étant donné l'absence d'un module dédié à cet effet dans l'architecture du dataspace. Des recommendations dans ce sens ont été formulées dans le rapport final soumis à la commission Européenne pour les futurs projets. L'usage du CSDM dans les différents sites pilotes a permis aux différents participants d'harmoniser leurs descriptions des données, optimisant ainsi l'intégration pour les fournisseurs de services, et augmentant la réutilisation des jeux de données pour les producteurs de données. Des travaux futurs sont prévus, pour offrir un outillage efficace à la question de la transformation des données brutes en graphes de connaissances. En dépit des technologies existantes, un public non averti pourrait bénéficier d'outils automatiques pour créer des mappings de schémas existants vers le CSDM, favorisant ainsi une meilleure utilisation.

Remerciements

Le projet OMEGA-X est financé par le programme Horizon Europe de l'UE dans le cadre de l'action d'innovation sous l'accord de subvention n° [101069287] (OMEGA-X).

Références

[1] Sarra Ben Abbes and Lynda Témal. D2.3 : Platoon common data models for energy version 2 - final. Technical Report D2.3, PLATOON, Mar 2022.

[2] Miracle Aniakor, Vinicius V Cogo, and Pedro M Ferreira. A survey on semantic modeling for building energy management. *arXiv preprint arXiv :2404.11716*, 2024.

[3] Drew Baigent, Mark Adamiak, Ralph Mackiewicz, and GMGM Sisco. Iec 61850 communication networks and systems in substations : An overview for users. *SISCO Systems*, 2004.

[4] Karima Boukir, Olivier Chaouy, and Bruno Traverson. Localised energy consumption aggregates using smart metering data. In *CIRED Workshop*, page Paper 0294, Ljubljana, Slovenia, June 2018. AIM.

[5] Serge Chávez-Feria, Raúl García-Castro, and María Poveda-Villalón. Chowlk : from uml-based ontology conceptualizations to owl. In *European Semantic Web Conference*, pages 338–352. Springer, 2022.

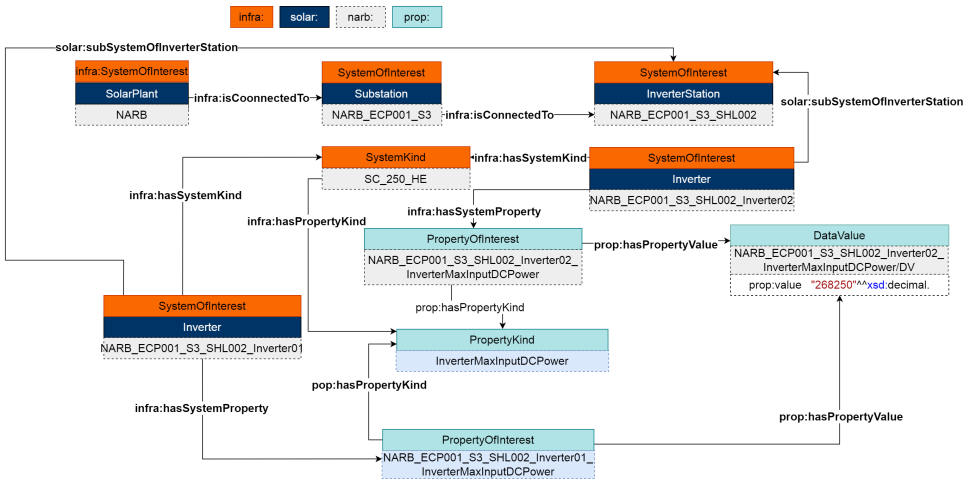

FIGURE 9 – Aperçu du diagramme CHOWLK de l'infrastructure du parc solaire de EDF Renouvelable -Narbonne-

[6] European Commission. Eg1 main report on interoperability and data access, 2019. Accessed : 2025-02-06.

[7] International Electrotechnical Commission. Iec 62325-351 ed.3 : Energy role ontology. Technical report, 2023. Accessed : 2025-02-06.

[8] International Electrotechnical Commission. 62746-4 demand side resource interface, 2024. Accessed : 2025-03-10.

[9] Laura Daniele, Cornelis Bouter, Georg Jung, Kristian Helmholt, Roderick van der Weert, Victor de Boer, and Carlos Manuel Pereira. D2.3 : Interoperable and secure standards and ontologies. Technical report, INTERCONNECT, Dec 2021.

[10] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. Rml : A generic language for integrated rdf mappings of heterogeneous data. *Ldow*, 1184, 2014.

[11] Stefan Feuerhahn, Michael Zillgith, Christof Wittwer, and Christian Wietfeld. Comparison of the communication protocols dlms/cosem, sml and iec 61850 for smart metering applications. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 410–415, 2011.

[12] Raúl García-Castro, Maxime Lefrançois, María Poveda-Villalón, and Laura Daniele. The etsi saref ontology for smart applications : a long path of development and evolution. *Energy Smart Appliances : Applications, Methodologies, and Challenges*, pages 183–215, 2023.

[13] Maliheh Haghgoo, Ilya Sychev, Antonello Monti, and Frank HP Fitzek. Sargon–smart energy domain ontology. *IET Smart Cities*, 2(4) :191–198, 2020.

[14] ISO. Iso 25012 the data quality model, 2008. Accessed : 2025-03-10.

[15] Maxime Lefrançois, Catherine Roussey, Fatma-Zohra Hannou, Victor Charpenay, and Antoine Zimmermann. Kind of x and x of interest : An ontology design pattern to reconcile web of thing ontologies. In *15th Workshop on Ontology Design and Patterns (WOP 2024)*, 2024.

[16] Maxime Lefrançois, Antoine Zimmermann, and Noorani Bakerally. A sparql extension for generating rdf from heterogeneous formats. In *The Semantic Web : 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14*, pages 35–50. Springer, 2017.

[17] Maxime Lefrançois, Jarmo Kalaoja, Takoua Ghariani, and Antoine Zimmermann. D2.2 seas knowledge model. Technical report, ARMINES-Fayol, 12 2016.

[18] Lina Nachabe, Fatma-Zohra Hannou, Maxime Lefrançois, and Marie Jubault. Toward agile interaction model based ontology development methodology (aime) for fair european data spaces. In *FOAM 2024 : FAIR principles for Ontologies and Metadata in Knowledge Management*, 2024.

[19] Boris Otto. A federated infrastructure for european data spaces. *Communications of the ACM*, 65(4) :44–45, 2022.

[20] RDFLib. Rdflib : A python library for working with rdf. `https://rdflib.readthedocs.io/en/stable/`, 2024. Accessed : 2024-06-01.

[21] Rafael Santodomingo, Mathias Uslar, Michael Specht, Sebastian Rohjans, Gareth Taylor, Stefan Pantea, Martin Bradley, Alan McMorran, et al. Iec 61970 for energy management system integration. *Smart Grid Handbook, 3 Volume Set*, 3 :375, 2016.

[22] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. The obo foundry : coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11) :1251–1255, 2007.

Méthode d'adaptation d'une ontologie d'application : cas des expérimentations agronomiques

Catherine Roussey[1], Anne Tireau[1], Pascal Neveu[1]

[1] MISTEA, Université de Montpellier, INRAE, Institut Agro, Montpellier

prenom.nom@inrae.fr

Résumé

Nous présentons une méthode pour adapter une ontologie d'application afin d'améliorer sa qualité pour qu'elle réponde aux exigences d'ingénierie des connaissances : inférence, alignement avec d'autres ontologies, cohérence, etc. Les ontologies d'application sont majoritairement utilisées dans les systèmes d'information et n'ont pas pour vocation à être partagées directement dans une large communauté. Nous avons ainsi identifié plusieurs activités pour adapter une ontologie d'application dont la principale est la construction/utilisation d'une ontologie du domaine. Nous avons appliqué notre méthode sur une ontologie créée il y a plus de 10 ans dans le domaine de la gestion des données d'expérimentations agronomiques.

Mots-clés

adaptation d'ontologie, retro-conception d'ontologie, restructuration d'ontologie, ingénierie directe d'ontologie, ontologie d'application, ontologie de domaine, ontologie d'expérimentations agronomiques.

Abstract

We present a method for reengineering an application ontology in order to improve its quality in order to meet current knowledge engineering requirements : inferences, ontologies alignment, coherency, ... Application ontologies are used in existing information systems and are not directly intended to be shared by a large community. We have therefore identified several activities to adapt an application ontology. The main one being the usage or the construction of a domain ontology. The domain ontology is aligned with reference ontologies. We applied our method to an ontology created over 10 years ago in the domain of agronomic experimentation data management.

Keywords

ontology reengineering, ontology reverse engineering, ontology restructuring, forward engineering ontology, application ontology, domain ontology, agronomic experimentation ontology.

1 Introduction

Une ontologie d'application offre le niveau de spécificité le plus fin, c'est-à-dire qu'elle est dédiée à un champ d'application précis à l'intérieur d'un domaine. Ainsi, elle décrit le rôle particulier des entités de l'ontologie de domaine dans ce champ. Par exemple, les spécifications d'un avion Airbus constitue une ontologie d'application précisant les concepts généraux pouvant provenir d'une ontologie de domaine de l'aéronautique. Une ontologie d'application est aussi un artefact sémantique développé pour répondre aux besoins d'une application [8]. Les ontologies d'application sont utilisées comme modèle de données pour les bases de graphes, impliquant ou non des inférences, ou comme modèle de données unifié pour interroger des sources de données hétérogènes [19]. Les ontologies d'application existent depuis le début des systèmes à base de connaissances, car elles constituent le cœur de ces systèmes. En revanche, elles sont souvent partagées uniquement dans la communauté utilisatrice, les conceptrices et concepteurs du système. Par conséquent, elles ne sont pas nécessairement publiées sur le Web. Dans nos travaux nous avons à notre disposition une ontologie d'application, formalisée dans les technologies du Web sémantique, qui existe depuis plus de 10 ans. Cette ontologie permet de gérer les données d'expérimentations agronomiques [12]. Elle n'intègre pas toutes les bonnes pratiques et toutes les dernières évolutions de la communauté d'ingénierie des connaissances : modularité des ontologies, alignement entre ontologies, publication sur le Web. Nous nous sommes engagés dans une démarche pour adapter cette ontologie d'application afin qu'elle intègre les éléments d'une ou plusieurs ontologies de domaine, qu'elle puisse être alignée avec d'autres ontologie de référence, et qu'elle permette de réaliser des inférences. Ainsi, nous espérons que la communauté utilisatrice tire rapidement profit de cet effort en structurant notre démarche. Dans ce sens, nous proposons une méthode capable d'adapter une ontologie pour améliorer sa qualité.

2 Méthodes d'adaptation d'ontologies

A notre connaissance il existe peu de travaux sur l'adaptation (reengineering) d'ontologies. Dans [6] l'adaptation d'une ontologie est définie comme le processus composé de 3 activités : 1) la rétro-conception (reverse engineering) reconstruit le modèle conceptuel d'une ontologie implémentée, 2) la restructuration transforme le modèle en

un nouveau modèle plus adéquat et 3) l'ingénierie directe (forward engineering) implémente le nouveau modèle dans un langage ontologique (identique ou différent de celui de l'ontologie initiale) pour produire une nouvelle ontologie. La méthode proposée a été utilisée pour adapter une ontologie des unités existantes en fonction d'un nouveau besoin [7]. Ce travail a été poursuivi dans la méthodologie Neon avec plusieurs scénarios : 4 «réutilisation et adaptation de ressources ontologiques existantes», 5 «réutilisation et fusion de ressources ontologiques existantes», 6 «réutilisation, fusion et adaptation de ressources ontologiques existantes»[17]. Dans les scénarios incluant l'adaptation, les 3 activités précédentes (rétro-conception, restructuration, ingénierie directe) sont toujours présentes. Des niveaux d'abstraction sont ensuite ajoutés : spécification, conceptualisation, formalisation, implémentation sur lesquels les changements vont être appliqués. Ce qui produit des parcours incluant les 3 activités appliquées sur chaque niveau d'abstraction sélectionné.

3 Méthode d'adaptation d'une ontologie d'application

Nous appliquons les étapes des activités présentées dans l'état de l'art, en ajoutant une activité de construction ou d'utilisation d'une ontologie de domaine. De plus, nous proposons de réorganiser les activités pour adapter une ontologie d'application. La différence étant que l'ontologie d'application est utilisée dans un système d'information en activité. Sa publication sur le Web est motivée par une volonté de transparence. De plus, nous souhaitons garder le système opérationnel pendant ce processus de ré-ingénierie. Notre approche se décompose en trois grandes activités schématisées dans la figure 1 :

- analyse de l'ontologie d'application et de sa documentation pour identifier les différents problèmes et à quel niveau d'abstraction ces difficultés sont apparues (spécification, conceptualisation, formalisation et implémentation).
- construction/utilisation d'une ontologie de domaine ayant une couverture similaire à celle de l'ontologie d'application. Ainsi elle servira de support pour faire évoluer et/ou corriger l'ontologie d'application de façon itérative.
- refonte de l'ontologie d'application en vérifiant que le système reste opérationnel, ce qui implique que notre utilisation de l'ontologie de domaine soit adaptée en fonction du comportement du système.

FIGURE 1 – Présentation générale de la méthode

3.1 Analyse de l'ontologie d'application

L'activité d'analyse correspond à l'activité de rétro-conception de l'état de l'art et se compose de 3 étapes comme présenté dans la figure 2.

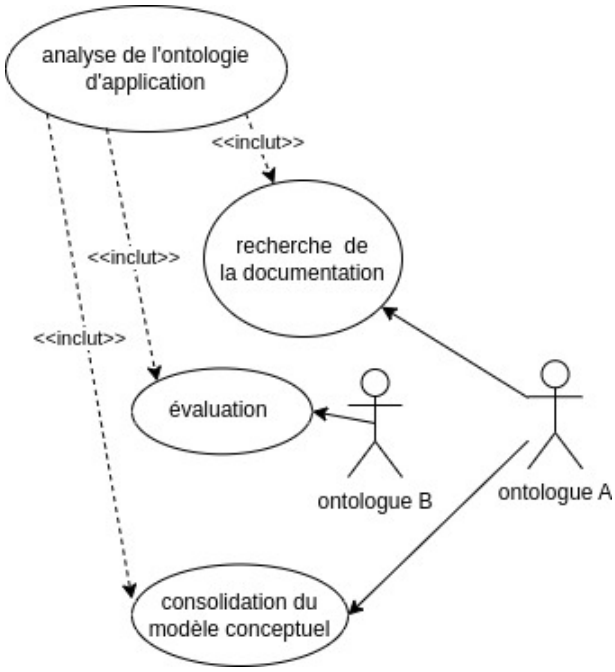

FIGURE 2 – Les étapes de l'analyse de l'ontologie d'application

A.1 Recherche de la documentation La documentation est constituée d'un ensemble de ressources incluant : les documents de spécification, les sources d'information utilisées au cours de la conception de l'ontologie, les modèles conceptuels et les différentes versions des fichiers sources (par exemple RDF/XML, OWL/XML).

A.2 Évaluation L'objectif de cette opération est d'identifier les évolutions nécessaires, les manques de l'ontologie à tous les niveaux (spécification, conception, formalisation, implémentation). Durant cette étape, des outils d'évaluation doivent être utilisés comme par exemple un raisonneur si le système nécessite des inférences ou d'autres outils d'évaluation de la qualité des ontologies comme OOPS [14] ou O'FAIRe [2]. Cette étape intègre aussi l'identification des bonnes pratiques de modélisation (les patrons de conception) et des mauvaises pratiques de modélisation (anti-pattern) présentes dans l'ontologie d'application. Le groupe de personnes en charge de cette étape doit être différent de l'équipe de conception initial de l'ontologie d'application.

A.3 Consolidation du modèle conceptuel L'objectif de cette étape est de concevoir, si il n'est pas disponible ou pas à jour dans la documentation, le modèle conceptuel actuel de l'ontologie d'application. Ainsi, il sera possible ultérieurement de le faire évoluer en fonction des résultats de l'étape précédente (cf C.4 et R.2). Dans cette étape, des outils de conception de diagramme doivent être utilisés, en préférant des outils d'édition collaborative.

3.2 Conception/utilisation de l'ontologie de domaine

L'ontologie de domaine est centrée sur la formalisation des connaissances du domaine, alors que l'ontologie d'application peut contenir des connaissances propres au fonctionnement du système, telles que les interactions avec les utilisatrices et utilisateurs. A ce titre, l'ontologie d'application utilisée dans un système n'a pas forcément vocation à être publiée sur le Web. Il se peut qu'elle ne soit pas une spécification directe d'une ontologie de domaine existante. Ce cas se produit plus spécifiquement lorsque l'ontologie d'application a été créée à partir d'une feuille blanche. Nous recommandons de construire une ontologie de domaine distincte de l'ontologie d'application. Ainsi, l'ontologie de domaine sera publiée sur le Web et servira de documentation pour l'ontologie d'application. Cette activité n'est pas identifiée directement dans l'état de l'art. L'équipe de conception de cette nouvelle ontologie peut intégrer les autrices et les auteurs initiaux de l'ontologie d'application ou de nouvelles personnes.

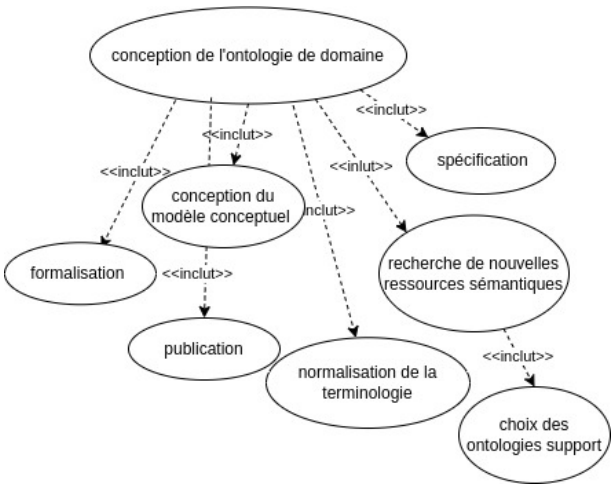

FIGURE 3 – Les étapes de la conception de l'ontologie de domaine

La figure 3 présente les étapes de cette activité. Cette activité reprend les étapes d'une méthode de conception d'ontologie comme Linked Open Term [13] ou Acimov [9] en prenant comme base les résultats de l'analyse de l'ontologie d'application précédente.

C.1 Spécification Les spécifications de l'ontologie d'application sont reformulées pour se concentrer sur les spécifications des connaissances du domaine (de quoi le système va parler) et supprimer les spécifications liées au besoin de l'application (interface graphique, gestion des utilisatrices et utilisateurs, etc…).

C.2 Recherche de ressources sémantiques Pour être à jour avec l'état des connaissances actuelles, une nouvelle recherche est effectuée pour identifier des ontologies du domaine proche, des ontologies noyau, des thésaurus, des terminologies voire des standards d'échange de données. L'accent est mis sur l'identification des patrons de conception à réutiliser en vue de corriger les mauvaises pratiques de modélisation identifiées précédemment.

C.3 Normalisation de la terminologie Cette étape sélectionne un terme, comme label préféré, non équivoque pour chaque élément de l'ontologie afin de lever les ambiguïtés identifiées dans l'analyse. Un élément de l'ontologie peut avoir plusieurs labels par langue mais un seul label préféré. Et inversement un label préféré ne peut être associé qu'à un seul élément de l'ontologie. Les ressources identifiées dans l'étape précédente sont utilisées pour poser des définitions en langue naturelle à chaque élément de l'ontologie du domaine. Les recommandations actuelles demandent une définition de type aristotélicienne, c'est à dire qui prend en compte le voisinage de l'élément. Une définition indique les points communs avec les parents et les différences avec les frères [1].

C.4 Conception du modèle conceptuel Au fur et à mesure de la normalisation, le modèle conceptuel initial de l'ontologie d'application est modifié pour représenter celui de l'ontologie de domaine en fonction des ajouts et des retraits. Cette étape est effectuée à l'aide d'un langage graphique et d'un outil dédié en suivant les questionnements de l'équipe de conception. Cette étape itérative se fait en collaboration avec l'équipe de conception du système d'information et leur communauté utilisatrice de ce système.

C.5 Formalisation Une fois le modèle conceptuel suffisamment avancé, l'ontologie de domaine est formalisée dans un des langages du Web sémantique (OWL par exemple). Le modèle est ainsi traduit à l'aide d'outils dédiés comme Protégé [11] ou à partir de fichiers tabulés qui seront transformés à l'aide d'outils comme Ontotext Refine [2] ou OpenRefine [3].

C.6 Publication sur le Web Cette étape inclut le fait de donner un identifiant pérenne à l'ontologie, de publier son code et sa documentation sur le Web à l'aide d'URIs déréférençables, de la rendre disponible dans un portail et de lui associer un dépôt GIT pour permettre à sa communauté utilisatrice de suivre son évolution et d'informer d'erreurs potentielles [4].

3.3 Refonte de l'ontologie d'application

La refonte a pour but de profiter des outils et des ressources de l'état de l'art, tout en améliorant de façon itérative l'ontologie d'application. Cette activité prend comme support l'ontologie de domaine, en lui ajoutant les éléments nécessaires au fonctionnement du système (notamment les interactions avec les utilisatrices et utilisateurs ou des agents logiciels). Cette activité est une évolution de l'activité d'ingénierie directe de l'état de l'art (voir section 2). Elle doit forcément impliquer les conceptrices et concepteurs initiaux

1. INRAE (2024), Rédiger une définition, quelques clés, Vocabulaires Ouverts@INRAE, https://vocabulaires-ouverts.inrae.fr/rediger-definition-quelques-cles/
2. https://www.ontotext.com/products/ontotext-refine/
3. https://openrefine.org/
4. https://vocabulaires-ouverts.inrae.fr/publier-vocabulaire/

de l'ontologie d'application. Il est préférable d'impliquer aussi l'équipe de conception de l'ontologie de domaine. La figure 4 présente les différentes étapes.

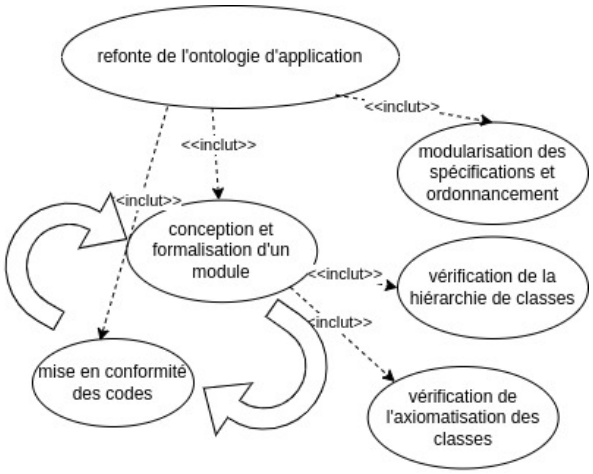

FIGURE 4 – Les étapes de la refonte de l'ontologie d'application

R.1 Modularisation des spécifications Cette étape réorganise les spécifications de l'ontologie d'application en module et ordonnancement des modules à la manière de la méthode Acimov [9].

R.2 Conception et formalisation d'un module Cette étape se concentre d'abord sur la conception et formalisation d'un module en travaillant d'abord sur la hiérarchie des classes. La relation hiérarchique a étudiée est d'abord la relation de subsomption comme `rdfs:subClassOf` puis les relations de méronymie comme `foaf:member` ou `dcterms:hasPart`. Cette étape inclut de vérifier la cohérence des définitions logiques des classes (axiomes) et l'alignement avec les classes des ontologies externes.

R.3 Mise en conformité des codes du système Cette étape implique de faire des tests unitaires du nouveau module puis de réaliser les tests d'intégration du module dans le système. Il se peut que les tests impliquent des changements dans le module qui doivent être reportés au niveau de l'ontologie de domaine. Comme le préconise Acimov, l'équipe itère ensuite sur le module suivant (itération de R.2 et R.3) jusqu'à la fin des spécifications.

4 Etude de cas sur l'ontologie OESO

PHIS est un système d'information piloté par des ontologies, qui est basé sur la suite logicielle OpenSILEX [12]. Il est conçu pour stocker, tracer, gérer et visualiser les données hétérogènes produites par les plateformes de phénotypage végétal à haut-débit : données phénotypiques (images, courbes de croissances des plantes, spectres, mesures manuelles), données environnementales (sol, atmosphère), données génétiques (taxonomie scientifique, lignées) et données sur la conduite des cultures (irrigations, traitements phyto, ...). Ces données sont hétérogènes sur plusieurs dimensions : plusieurs échelles structurelles (du gène à la parcelle) et temporelles (mesures à la minute, mesures annuelles) provenant d'environnements variés (champ, serre, exploitation agricole). Pour gérer ces données, l'architecture OpenSILEX combine une gestion des données de flux par un système NoSQL (MongoDB), avec une gestion des données «statiques»par une base de graphes (RDF4J). Les données de flux ont besoin d'être contextualisées par des métadonnées décrivant les ressources biologiques, les capteurs, les plateformes, les types de mesures, ... L'ontologie d'application qui structure la base de graphes est intitulée «Experimental Scientific Objects Ontology»(OESO). Elle a été créée en 2012 en s'inspirant des éléments de l'ontologie des expérimentations scientifiques EXPO [16]. EXPO est une ontologie générique datant de 2006 qui utilise SUMO. EXPO n'est pas disponible en téléchargement. Elle n'est pas orientée sur la structuration des données mais plus sur la documentation d'une expérimentation. Notre objectif est de remettre à jour OESO en fonction des nouvelles ontologies.

Actuellement, nous avons déjà réalisé l'activité d'analyse de l'ontologie OESO et commencé l'activité de conception de l'ontologie de domaine. Les résultats seront disponibles sur un dépôt GIT.

4.1 Analyse de OESO

Plusieurs versions de OESO sont disponibles dans des répertoires GIT privés. Initialement, l'ontologie s'intitulait Ontology For Experimental Phenotyping Object (OEPO) [5]. La dernière version (1.2.1) de OESO est visible sur WebProtégé pour permettre aux développeuses et développeurs de proposer des modifications. Nous avons centré notre analyse sur cette version.

A.1 Recherche de la documentation Aucun document ne décrit l'ensemble des spécifications de OESO.

Différents rapports (livrables de projet, thèse, rapports de stage...) contiennent des sous-parties du modèle conceptuel de OESO. Un modèle conceptuel complet combinant les éléments NoSQL et les éléments de OESO a été construit à l'aide de draw.io en suivant le formalisme de WebVOWL. Il a été diffusé sous un format poster.

OESO est formalisée en OWL. OESO n'importe aucune ontologie directement, en revanche elle utilise des éléments provenant d'autres ontologies (PROV, FOAF, OA, Time, Vcard, DCTERMS, SKOS, ORG). OESO se compose de 126 classes, 60 propriétés objets, 87 propriétés de type de données, 16 propriétés d'annotations et 160 individus. Elle inclut deux modules Ontology of Experimental Events (OOEV) et security. Le module OEEV définit une hiérarchie des évènements. Auparavant ce module existait comme ontologie à part entière dans les versions précédentes de OESO. Enfin, le module security permet de gérer les droits des utilisatrices et utilisateurs.

A.2 Evaluation Pour analyser OESO nous avons utilisé l'outil OOPS. Comme le montre la Figure 5 : 10 types d'écueils (Pitfalls) différents ont été détectés par cet outil. Les plus importants en nombre sont : les 216 éléments de

5. `https://agroportal.lirmm.fr/ontologies/OEPO`

l'ontologie non documentés P08 ; les 55 propriétés objets définies sans domaine et codomaine. Ces propriétés objets n'ont pas de propriétés inverses définies. Les écueils les plus intéressants concernent les 2 classes qui ont le même label et les classes équivalentes détectées.

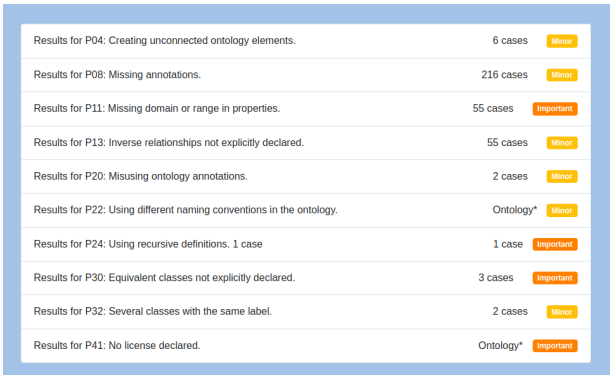

FIGURE 5 – Résultat de l'évaluation de OOPS

Une personne experte en ingénierie des connaissances a aussi parcouru l'ensemble de l'ontologie manuellement pour détecter les différents écueils. Cette ontologie ayant été développée par un collectif sur le long terme, il est compréhensible de trouver des incohérences telles que : P02 l'existence de classes ayant des labels synonymes, P07 la fusion de concepts dans une même classe. 7 écueils non mentionnés dans la littérature ont été détectés.

- L'utilisation d'un patron de conception structurel du développement logiciel (le patron composite), qui induit de mauvaises propriétés logiques s'il est ainsi formalisé.
- L'utilisation des éléments de syntaxe RDF comme élément de l'ontologie. Par exemple, la classe `rdf:Bag` a été utilisée pour représenter des collections d'entités.
- L'utilisation de propriété d'annotation dans la déclaration de contraintes. Par exemple, la partie d'axiome `<label min 1 xsd:string>` exprime la contrainte qu'un label soit toujours associé à un élément de l'ontologie donné.
- Le mauvais usage de la relation de spécialisation entre classes (`rdfs:subClassOf`) à la place d'une relation de méronymie (composition).
- L'absence de classes soeurs dans une hiérarchie incomplète, ou la réorganisation de la hiérarchie. Par exemple, la hiérarchie des documents (`oeso:Document`) contient des classes `oeso:Multimedia`, `oeso:InteractiveResource`, `oeso:Dataset` et `oeso:WebSite`. La hiérarchie des fichiers (`oeso:Datafile`) contient les classes `oeso:Image` et `oeso:Archive`.
- A l'inverse des classes ont des labels laissant imaginer qu'elles ont un lien entre elles qui n'existent pas actuellement dans l'ontologie. Par exemple, la classe `oeso:Group` n'a aucun lien avec les classes `oeso:VariablesGroup` et `oeso:GermplamsGroup`.
- Certains éléments d'ontologie externe utilisés dans OESO sont obsolètes.

En résumé, des relations de méronymie entre classes ont été mal formalisées dans OESO. Un autre problème vient du fait de la réutilisation d'éléments venant d'ontologies externes ayant des labels similaires (`foaf:Agent` et `prov:Agent`). Ceci s'explique par le fait que les conceptrices et concepteurs ont voulu inclure différentes ontologies au cours du temps sans vérifier la cohérence globale.
L'évaluation manuelle a aussi permis de détecter un nouveau patron de conception ontologique : le patron liant un organisme vivant à sa description taxonomique et/ou génétique décrite dans une ressource génétique (GermBank).
Pour finaliser, chaque élément de l'ontologie a été annoté sur WebProtégé pour indiquer son statut : à corriger, en cours ou valide. La figure 6 montre un extrait de cette étape.

A.3 Consolidation du modèle conceptuel Le modèle construit à l'aide de draw.io correspond à la version actuelle de OESO. Vu sa taille importante, il est envisagé de le représenter en plusieurs sous-parties orientées sur un besoin explicite.

4.2 Conception de l'ontologie de domaine

Nous allons définir une ontologie du domaine de l'expérimentation agronomique intitulée «Experiment on Living Organisms in Agriculture»(ELOA).

C.1 Spécification Vu l'absence de spécification actuelle disponible, une série de questions de compétences sont en cours de rédaction pour mieux définir les besoins de ELOA. Ces spécifications enrichissent les questions de compétences de REPRODUCE-ME [15]. Les besoins et les éléments de OESO qui sont en lien avec le fonctionnement du système PHIS (la gestion des utilisatrices et utilisateurs, la gestion des interfaces et des fichiers NoSQL) ont été identifiés et supprimés.

C.2 Recherche de ressources sémantiques Depuis quelques années, plusieurs ontologies du domaine de l'expérimentation, en général, et de l'expérimentation agronomique en particulier, ont vu le jour. Dans l'écosystème de BFO, nous pouvons citer :

- Experimental Factor Ontology (EFO)[6] est utilisé pour annoter les types de variables scientifiques présents dans la base EBO.
- Ontology for Biomedical Investigations (OBI) [4] décrit les études cliniques et biomédicales.
- Agronomy Ontology (AgrO)[7] est utilisé pour annoter les tableaux constituant les carnets de terrains des expérimentations de semenciers.

Dans l'écosystème de PROV, nous avons des ontologies plus génériques du domaine de l'expérimentation : P-Plan [5] étend Prov pour décrire les méthodes et les variables utilisées comme entrée/sortie de ces méthodes. L'ontologie «workflow description vocabulary»(wfdesc) intègre les

6. `https://www.ebi.ac.uk/efo/`
7. `https://agroportal.lirmm.fr/ontologies/AGRO`

ontologies Object Reuse and Exchange (ORE), Annotation Ontology (AO) et PROV pour décrire les traitements des données scientifiques [3]. REPRODUCE-ME [15] intègre PROV, P-Plan et la Time ontologie pour décrire les expérimentations, leurs protocoles et des données d'entrées/sorties. Plus récemment un nouveau patron de conception ontologique sur les expérimentations a été proposé dans [10]. Dans le domaine des expérimentations agronomiques il existe plusieurs standards d'échange de données. Le plus connu est le format ICASA [18]. Le modèle de données décrit dans ISA spécification est aussi très utilisé. Enfin il existe un standard décrivant les bonnes pratiques de conduites des expérimentations des produits phytosanitaires produit par Organisation Européenne et Méditerranéenne pour la Protection des Plantes (OEPP/EPPO) [1].

C.3 Normalisation de la terminologie Comme indiqué dans l'étape d'évaluation, plusieurs termes synonymes ou proches ont été utilisés pour définir des classes qui n'ont pas de liens entre elles. Un travail de normalisation et de définition en langue naturelle est en cours. Il s'est avéré qu'après discussions avec les développeuses et développeurs, les classes qui partagent des labels identiques ont en fait des usages bien différents. Par exemple, le `foaf:Agent` est utilisé pour décrire les utilisatrices et utilisateurs du système PHIS, alors que le `prov:Agent` décrit les personnes impliquées dans les expérimentations agronomiques.

C.4 Conception du modèle conceptuel Nous avons décidé de spécialiser l'ontologie PROV pour décrire les expérimentations agronomiques. A partir du texte du standard de OEPP/EPPO, et des terminologies extraites de ICASA et ISA spécification nous allons définir les classes et spécifier leur lien de spécialisation avec les classes de PROV. Nous allons aussi essayer de nous appuyer sur l'ontologie AgrO (en utilisant des alignements SKOS) pour conserver une compatibilité avec BFO.

5 Discussion et conclusion

Une ontologie d'application en production est développée avec une pression constante de la communauté utilisatrice. Ce type d'ontologie, par exemple OESO, est développé par des collectifs ayant souvent des niveaux de compétence en ingénierie des connaissances hétérogènes. Compte tenu des deux points cités précédemment il est évident qu'un collectif exploitant des ressources ontologiques pour une application s'expose à des risques importants d'apparition d'écueils tout au long de sa vie. Ainsi nous avons proposé une méthode d'adaptation de l'ontologie d'application pour détecter et gérer ces difficultés.
Un des problèmes les plus difficiles à gérer dans une ontologie d'application en cours d'exploitation est de gérer en continue les évolutions des ontologies externes réutilisées. Nous n'avons pas encore trouver une solution acceptable.

Remerciements

Les auteurs tiennent à remercier l'ensemble de l'équipe OpenSILEX. Ces travaux ont été financé en partie par le projet CASDAR Connaissances 2024 «Standardiser les données expérimentales et techniques pour faciliter leur réutilisation et accélérer l'innovation et le développement agricole : application aux travaux sur les biosolutions»(STAR) financé par FranceAgriMer et le projet ANR «Infrastructure Biologie Santé» PHENOME-EMPHASIS project (ANR-11-INBS-0012) financé par l'Agence Nationale de la Recherche et le Programme d'Investissements d'Avenir (PIA).

Références

[1] Pp1/152(4) : Design and analysis of efficacy evaluation trials. Technical Report 3, OEPP/EPPO, 2012.

[2] Emna Amdouni, Syphax Bouazzouni, and Clement Jonquet. O'faire makes you an offer : metadata-based automatic fairness assessment for ontologies and semantic resources. *International Journal of Metadata, Semantics and Ontologies*, 16(1) :16–46, 2022.

[3] Khalid Belhajjame, Jun Zhao, Daniel Garijo, Matthew Gamble, Kristina Hettne, Raul Palma, Eleni Mina, Oscar Corcho, José Manuel Gómez-Pérez, Sean Bechhofer, et al. Using a suite of ontologies for preserving workflow-centric research objects. *Journal of Web Semantics*, 32 :16–42, 2015.

[4] Ryan R Brinkman, Mélanie Courtot, Dirk Derom, Jennifer M Fostel, Yongqun He, Phillip Lord, James Malone, Helen Parkinson, Bjoern Peters, Philippe Rocca-Serra, et al. Modeling biomedical experimental processes with obi. In *Journal of biomedical semantics*, volume 1, pages 1–11. Springer, 2010.

[5] Daniel Garijo Verdejo and Yolanda Gil. Augmenting prov with plans in p-plan : scientific processes as linked data. CEUR Workshop Proceedings, 2012.

[6] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering : with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media, 2006.

[7] Asunción Gómez-Pérez and Ma Dolores Rojas-Amaya. Ontological reengineering for reuse. In Dieter Fensel and Rudi Studer, editors, *Knowledge Acquisition, Modeling and Management*, pages 139–156, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[8] Nicola Guarino. Understanding, building and using ontologies. *International journal of human-computer studies*, 46(2-3) :293–310, 1997.

[9] Fatma-Zohra Hannou, Victor Charpenay, Maxime Lefrançois, Catherine Roussey, Antoine Zimmermann, and Fabien Gandon. The acimov methodology : agile and continuous integration for modular ontologies and vocabularies. In *MK 2023-2nd Workshop on Modular Knowledge associated with FOIS 2023-the 13th International Conference on Formal Ontology in Information Systems*, 2023.

[10] Jacques Hilbey, Xavier Aimé, and Jean Charlet. Un patron de conception pour la modélisation ontologique des paradigmes expérimentaux. In *34es*

Journées francophones d'Ingénierie des Connaissances (IC 2023)@ Plate-Forme Intelligence Artificielle (PFIA 2023), pages 28–33, 2023.

[11] Mark A Musen. The protégé project : a look back and a look forward. *AI matters*, 1(4) :4–12, 2015.

[12] Pascal Neveu, Anne Tireau, Nadine Hilgert, Vincent Nègre, Jonathan Mineau-Cesari, Nicolas Brichet, Romain Chapuis, Isabelle Sanchez, Cyril Pommier, Brigitte Charnomordic, et al. Dealing with multi-source and multi-scale information in plant phenomics : the ontology-driven phenotyping hybrid information system. *New Phytologist*, 221(1) :588–601, 2019.

[13] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. Lot : An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111 :104755, 2022.

[14] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. Oops !(ontology pitfall scanner !) : An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2) :7–34, 2014.

[15] Sheeba Samuel and Birgitta König-Ries. Reproduceme : ontology-based data access for reproducibility of microscopy experiments. In *The Semantic Web : ESWC 2017 Satellite Events : ESWC 2017 Satellite Events, Portorož, Slovenia, May 28–June 1, 2017, Revised Selected Papers 14*, pages 17–20. Springer, 2017.

[16] Larisa N Soldatova and Ross D King. An ontology of scientific experiments. *Journal of the royal society interface*, 3(11) :795–803, 2006.

[17] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, and Aldo Gangemi. Introduction : Ontology engineering in a networked world. In *Ontology engineering in a networked world*, pages 1–6. Springer, 2011.

[18] Jeffrey W White, LA Hunt, Kenneth J Boote, James W Jones, Jawoo Koo, Soonho Kim, Cheryl H Porter, Paul W Wilkens, and Gerrit Hoogenboom. Integrated description of agricultural field experiments and production : The icasa version 2.0 data standards. *Computers and electronics in agriculture*, 96 :1–12, 2013.

[19] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access : A survey. pages 5511–5519. International Joint Conferences on Artificial Intelligence, 2018.

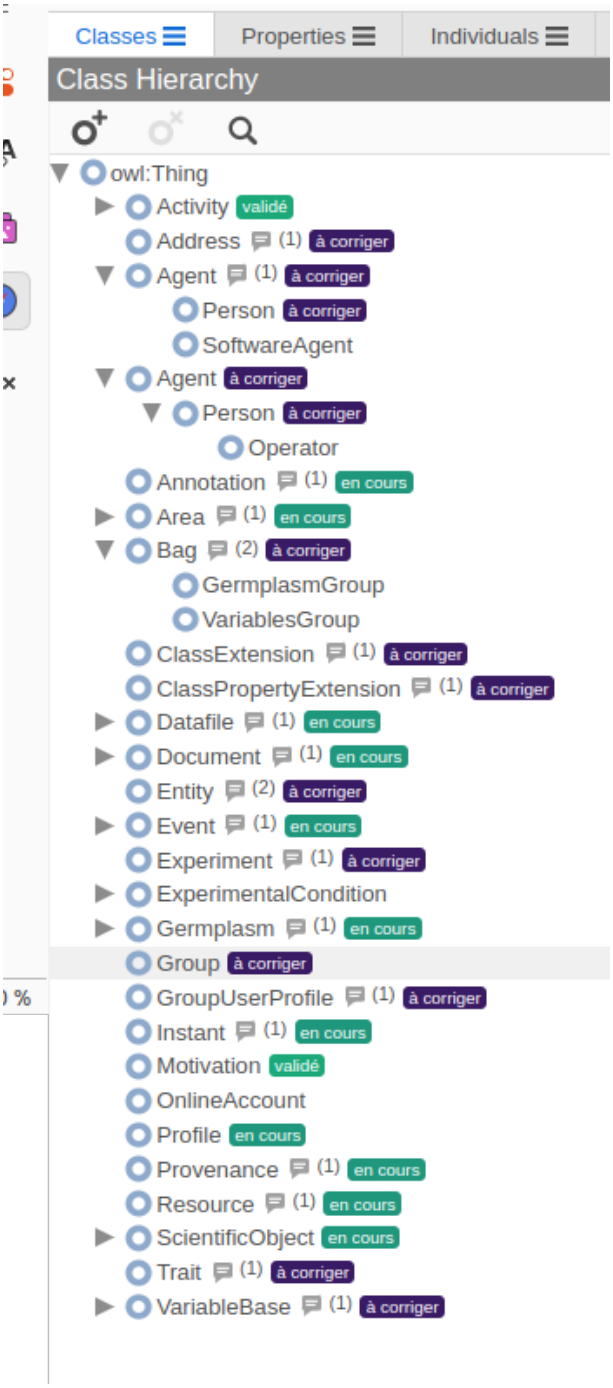

FIGURE 6 – Extrait de OESO sur WebProtégé

OntoPFAS : Ontologie des PFAS et de leur exposition

Davide Di Pierro[1], Lylia Abrouk[1,2], Alexis Guyot[3],
Danai Symeonidou[1], Pierre Labadie[4], Benjamin Lysaniuk[5]

[1] MISTEA, INRAE & Institut Agro, France
[2] LIB, Université Bourgogne Europe, France
[3] LIRMM, Université de Montpellier, CNRS, France
[4] CNRS UMR EPOC, Talence
[5] CNRS UMR PRODIG, Aubervilliers

davide.di-pierro@umontpellier.fr

Résumé

La construction d'ontologies est une des premières tâches dans le domaine de la représentation des connaissances. Elle reste très pertinente aujourd'hui, grâce à l'expressivité des langages formels, qui permettent encore d'explorer et de découvrir des connaissances. Pendant des décennies, la communauté a développé des méthodologies pour la construction manuelle d'ontologies, et plusieurs classifications de celles-ci ont été proposées. Dans cet article, nous présentons une méthodologie de construction d'ontologie basée sur des méthodes existantes, et nous l'appliquons à la représentation du domaine des PFAS (Per- et poly- fluoroalkyle substances) et de leur exposition. Les PFAS sont des substances dont la structure chimique particulière les rend très résistantes et efficaces dans de nombreuses applications industrielles. Ils suscitent un intérêt croissant en raison de leur impact négatif sur la santé et l'environnement. Ce travail s'inscrit dans le cadre du projet interdisciplinaire DAE (Détection d'Anomalies Environnementales).

Mots-clés

Ontologie, représentation des connaissances, PFAS, environnement.

Abstract

Ontology construction is one of the earliest tasks in the field of knowledge representation. It remains highly relevant today, thanks to the expressiveness of formal languages, which still allow for the exploration and discovery of knowledge. For decades, the community has developed methodologies for the manual construction of ontologies, and several classifications of these have been proposed. In this article, we present a methodology for constructing ontologies based on existing methods, and we apply it to represent the domain of PFAS and their exposition. PFAS are substances with a unique chemical structure that makes them highly resistant and effective in many industrial applications. They have been drawing increasing attention due to their negative impact on health and the environment. This work is part of the interdisciplinary DAE (Environmental Anomaly Detection) project.

Keywords

Ontology, knowledge representation, PFAS, environment.

1 Introduction

La représentation des connaissances et l'ingénierie des ontologies sont parmi les premières thématiques de recherche en intelligence artificielle, et elles continuent d'attirer une attention, tant sur le plan théorique que pratique. Une ontologie constitue une représentation formelle des éléments qui composent un domaine spécifique [20]. Grâce à l'expressivité des langages formels utilisés pour décrire ces ontologies, il est possible d'introduire un raisonnement logique permettant d'inférer de nouvelles connaissances ou de détecter des incohérences dans les informations existantes. Bien que les ontologies aient trouvé des applications dans divers domaines, aucune conceptualisation des connaissances relatives aux PFAS n'a encore été élaborée.

Les substances per- et polyfluoroalkyles (PFAS) sont des composés organiques fluorés possédant des propriétés physiques, chimiques et biologiques très diverses [2]. D'un point de vue socio-économique, les PFAS répondent à de nombreux besoins et sont utilisés dans une variété de contextes industriels. Cependant, leur persistance dans l'environnement et leur toxicité ont un impact négatif sur la santé humaine et l'écosystème. Plusieurs études récentes ont exploré ces effets, notamment en analysant les concentrations de PFAS et leur devenir dans les masses d'eau, l'une des principales voies d'exposition humaine [12, 19].

Face à l'urgence d'analyser et de répondre à diverses questions concernant les PFAS, il est crucial de développer une classification formelle de ces substances et de leurs interactions avec l'environnement. Bien que certaines classifications informelles existent, aucune n'a été conçue pour spécifier la présence des PFAS dans l'environnement, un facteur essentiel pour comprendre le phénomène et réguler leur utilisation. La classification que nous proposons vise deux objectifs principaux : d'une part, offrir une ressource ouverte qui puisse être utilisée et enrichie par la communauté

scientifique ; d'autre part, permettre l'intégration de cette classification avec des données ou ressources existantes afin de découvrir de nouvelles connaissances sur les PFAS et le phénomène dans son ensemble, en exploitant les mécanismes de raisonnement pour identifier des incohérences ou des comportements atypiques.

Dans le cadre du projet *Forever Pollution* [5], le journal *Le Monde*, a publié une carte de la pollution éternelle [1], recensant les coordonnées géographiques des sites pollués ainsi que des informations détaillées sur les échantillons de PFAS prélevés. Nous exploiterons les données disponibles afin d'enrichir et peupler notre ontologie. En janvier 2024, le CNRS a lancé le projet PDH [2], dont l'objectif est d'enrichir ces données, d'en assurer l'actualisation continue et de développer des solutions durables. Dans ce contexte, notre projet DAE *(Détection d'anomalies environnementales)*, s'appuie sur ces données comme base et mobilise des compétences interdisciplinaires en sciences des données, chimie environnementale et géographie.

La suite de cet article est organisée de la manière suivante : dans la section 2, nous présentons l'état de l'art sur les méthodes de construction d'ontologies et les ontologies existantes liées aux PFAS. La section 3 décrit notre approche pour modéliser les PFAS à travers une ontologie, tout en répondant aux questions de compétence et en évaluant notre travail selon les principes FAIR. Enfin, la section 4 conclut l'article et explore les extensions possibles de ce travail.

2 Travaux antérieurs

Méthodes pour la construction d'ontologies : En raison de la diversité des domaines et des projets nécessitant leur élaboration, une riche littérature a émergé sur les différentes méthodes et approches. Ces méthodes peuvent être classifiées selon plusieurs critères, tels que la généralité de leur stratégie, l'approche adoptée (ascendante ou descendante), et leur niveau de formalisation [6, 18]. Parmi les approches les plus établies et reconnues figurent *Methontology* [10] et *Ontology Development 101* [13]. La première, qui est l'une des approches les plus anciennes, souffre d'un inconvénient majeur : une phase d'intégration qui intervient trop tardivement, après la formalisation, et l'absence d'une étape itérative pour l'amélioration. En revanche, la seconde anticipe mieux l'intégration avec d'autres ressources et propose une séquence de documents préparés avant la formalisation, en retardant autant que possible l'utilisation du langage formel.

Certaines approches mettent l'accent sur la séparation des rôles et des responsabilités dans la construction de l'ontologie. Par exemple, l'approche $(KA)^2$ [1] divise le processus en deux groupes distincts : le groupe de représentation des connaissances et le groupe de contrôle. Bien que cette approche soit généralement applicable, elle est particulièrement recommandée dans les domaines techniques (ex : médical) en raison de sa rigueur et de sa spécialisation. L'une des approches distribuées les plus connues est DILIGENT [15], proposée comme une méthodologie qui commence par le travail conjoint d'experts en ontologie et en domaine, et qui conduit ensuite à des raffinements successifs fournis par les utilisateurs qui commencent à travailler avec l'ontologie. Un groupe CORE est ensuite chargé de fusionner les raffinements locaux avec la conceptualisation partagée, si nécessaire. Dans notre cas, la différence entre les utilisateurs et le groupe CORE ne serait pas si évidente, et l'introduction de nouveaux domaines rend les étapes itératives nécessaires.

Les stratégies ultérieures ont tenté d'atténuer les inconvénients d'une approche ontologique en cascade en utilisant les principes agiles. Citons par exemple SAMOD [14] (ou son extension ACIMOV [11]) pour gérer correctement les changements d'élicitations et la coordination entre différents groupes en développant des *modelet* de scénarios. UPONLite [7] suit les étapes traditionnelles identifiées par la méthodologie de la chute d'eau, en introduisant des itérations dans toutes les phases après la définition du glossaire. L'interaction entre les utilisateurs et les experts est facilitée par des interfaces visuelles. Dans le contexte des méthodes agiles, la communication entre les personnes impliquées doit être accélérée. C'est pourquoi les approches actuelles comme AgiSCOnt [21] visent principalement à faire parler le même langage aux non-experts et aux experts par le biais d'interfaces visuelles et de cas d'utilisation. Ces derniers guident le développement de nouvelles versions.

Avec l'essor des données liées, il devient nécessaire pour les ontologies de s'adapter à cette nouvelle source d'informations. La méthodologie *Linked Open Terms* (LOT) [17] se distingue par sa généralité, sa formalité logicielle et son caractère itératif. Par rapport à de nombreuses autres méthodologies, elle est plus exhaustive et fournit des lignes directrices pour structurer efficacement les connaissances. De plus, elle peut être adoptée à différents niveaux de formalité et comprend une partie d'intégration pour faciliter la réutilisation.

Ontologies existantes : La première classification des PFAS a été proposée par Buck et al. en 2011 [2], et se concentre principalement sur la structure chimique des substances. La classification la plus récente et la plus complète des PFAS a également été proposée par Buck et al. en 2021 [3]. Bien qu'elle soit encore largement axée sur l'aspect chimique, elle inclut désormais des informations supplémentaires sur la présence des PFAS et leurs différentes sous-catégories. Bien qu'aucune ontologie ne soit spécifiquement dédiée aux PFAS et à leur impact environnemental, plusieurs ontologies et vocabulaires connexes existent. Les connaissances qu'ils encapsulent peuvent être réutilisées et adaptées à notre contexte.

Dans le domaine de la chimie, l'ontologie ChEBI [8] est une référence, couvrant une large gamme de substances chimiques et bénéficiant d'un consensus général au sein de la communauté scientifique. D'autres ressources pertinentes

1. https://www.lemonde.fr/en/les-decodeurs/article/2023/02/23/forever-pollution-explore-the-map-of-europe-s-pfas-contamination_6016905_8.html
2. https://pdh.cnrs.fr/en/

pour nos objectifs incluent l'Exposure Ontology (ExO) [3] et la terminologie Tox21 [4]. L'ontologie ExO se concentre sur l'exposition humaine à des substances environnementales affectant la santé. Elle couvre des aspects tels que les maladies, les réactions physiologiques, l'anatomie humaine et les principaux phénomènes d'exposition, tout en prenant en compte les dimensions temporelles et spatiales. De son côté, Tox21 aborde la toxicité, les substances toxiques et leurs effets sur la santé.
L'ontologie ENVO [5] constitue une référence pour la modélisation du domaine environnemental. Elle vise à fournir des vocabulairespermettant de caractériser aussi bien les environnements naturels que ceux modifiés par l'activité humaine, ainsi que les processus biologiques et leurs interactions. Par ailleurs, les concepts de l'ontologie SOSA [6] peuvent être réutilisés pour la représentation des mesures.
Nous nous appuyons sur ces ressources pour élaborer une première version de notre ontologie dédiée aux PFAS et à leur impact sur l'environnement.

3 Approche

Dans cette section, nous présentons notre méthodologie pour la modélisation de l'ontologie *OntoPFAS* développé dans le cadre de notre projet, qui vise à modéliser et structurer les relations entre les PFAS et leur exposition. Nous avons basé notre approche sur la méthodologie *Linked Open Terms*, en l'adaptant afin d'avoir une approche itérative avec plusieurs groupes d'experts.
L'objectif est de proposer une méthode applicable dans divers contextes dans lesquels de nombreux groupes d'experts collaborent.Aussi, toutes les exigences ne sont pas explicites dès le départ, en raison du manque de travaux ou de données existantes ; une évaluation experte est requise pour appréhender la complexité et la nature scientifique du domaine.
Dans un premier temps, nous introduisons les questions de compétence auxquelles l'ontologie devra répondre. Les premières questions identifiées avec les experts (un géographe et un chimiste) sont les suivantes :

1. Quels sont les PFAS et comment sont-ils classés en fonction de leur structure chimique ?
2. Quelles sont les principales sources industrielles et naturelles des PFAS ?
3. Quels processus environnementaux influencent la dispersion des PFAS dans les différents compartiments (air, eau, sol, biote) ?
4. Quelles sont les méthodes analytiques permettant de détecter et quantifier les PFAS ?
5. Quels sont les sites où les PFAS sont les plus présents ?
6. Comment la présence de PFAS sur un site évolue-t-elle dans le temps ?
7. Quels sont les PFAS les plus mesurés ?

La figure 1 illustre notre méthodologie. Les phases entourées d'une bordure noire (Fusion de glossaires et Validation des modifications) représentent les deux éléments ajoutés dans la méthodologie LOT.
Dans la *Fusion de glossaire* phase, l'objectif est de consolider les concepts communs à plusieurs domaines et d'en fournir une définition cohérente et complète pour l'ensemble des disciplines concernées. Cette étape influence également la réutilisation des ontologies existantes, en guidant le choix vers les sources les plus pertinentes et les plus complètes. Quant à la *Validation des modifications*, elle nécessite une évaluation rigoureuse des changements par des experts dans chaque domaine. Ce processus est itératif, car tout changement peut déclencher une cascade d'ajustements successifs. Cette phase peut également se produire lorsqu'un domaine entièrement nouveau doit être introduit, ce qui entraîne la nécessité d'ajouter de nouvelles exigences à celles qui existent déjà dans les autres domaines. Les phases ajoutées sont menées avec l'aide d'experts du domaine qui veillent à ce que les exigences (resp. les changements) soient conformes à leur domaine.

3.1 Construction d'ontologie

Dans cette section, nous proposons notre première version de l'ontologie *OntoPFAS*. Le préfixe de notre ontologie est « **ontopfas** ». Nous présentons ici la liste des principaux termes qui représentent les différentes aspects de cette première conceptualisation : **PFAS** suit la définition de Buck, définie du point de vue chimique plutôt que de son impact sur la santé et l'environnement. L'ontologie ChEBI contient le concept équivalent [7].
Étant donné l'existence du concept PFAS dans l'ontologie, nous le réutilisons et commençons notre conceptualisation des sous-classes de PFAS à partir de ce concept. Nous n'énumérons ici que certaines des sous-classes telles que **Perfluoroalkyl**, **Polyfluoroalkyl**, **Long chain**, **Short chain**. Nous ne fournissons pas de détails chimiques, mais ces classes donnent des indications sur la toxicité et la résistance des composés PFAS. **Functional group** représente une classe pertinente, indiquant le groupe fonctionnel du PFAS. Il s'agit du composé chimique responsable de la toxicité. Les PFAS contenant du carbone, le concept **Carbon** doit être représenté. Plus précisément, ses instances correspondent au nombre d'atomes de carbone présents dans le composé, comme C_6 ou C_8. Les PFAS à chaîne longue et courte peuvent être déduits du nombre d'atomes de carbone. **Contamination location** indique le point géographique de la présence d'un PFAS. Un concept essentiel est **Observation**, qui recueille des informations sur la mesure d'un PFAS à un endroit donné. La mesure conserve également la trace de la source (ensemble de données) à laquelle cette mesure appartient (**Dataset**). L'origine peut être exprimée à l'aide des propriétés du Dublin Core. Ces propriétés sont essentielles, notamment pour indiquer la source d'un jeu de données concernant une mesure de PFAS dans

3. https://bioportal.bioontology.org/ontologies/EXO
4. https://tox21.gov/overview/
5. https://bioportal.bioontology.org/ontologies/ENVO
6. https://www.w3.org/TR/vocab-ssn/
7. http://purl.obolibrary.org/obo/CHEBI_172397

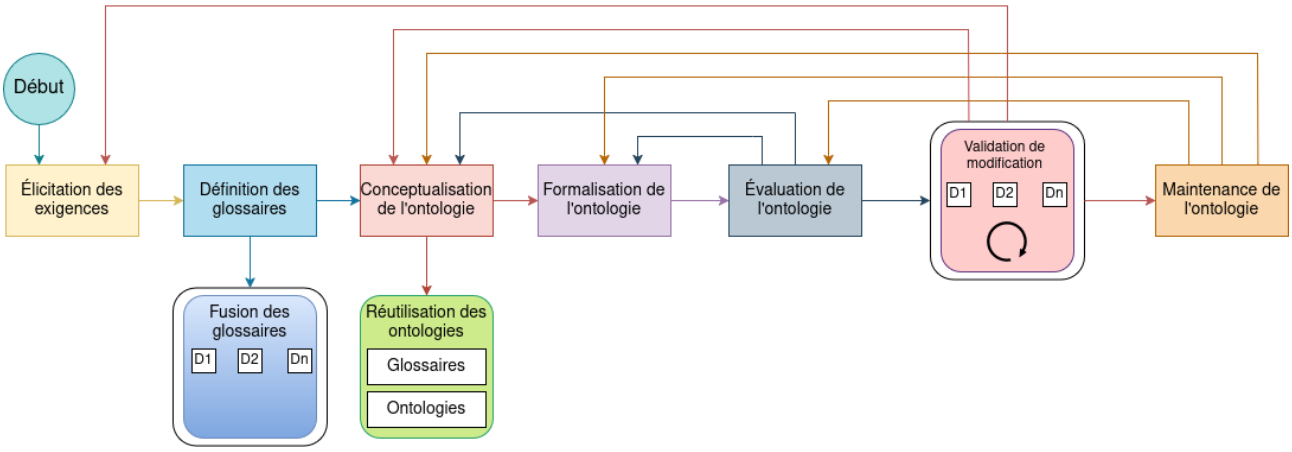

FIGURE 1 – Notre méthodologie à partir de LOT

l'environnement. La mesure est également associée à une **Occupation**, qui indique les caractéristiques géographiques du lieu où la mesure a été effectuée. Les instances de cette classe sont associées à un **Environment** (eau, sol, etc.). Les PFAS sont à la fois utilisés et générés par diverses activités, représentées par **Activity**, qui se spécialisent selon les secteurs d'application, qu'ils soient privés ou publics. L'exposition est ensuite modélisée par **Exposure**, qui intègre des informations sur un PFAS affectant certaines personnes (**Human individual**). Parmi elles, les individus vulnérables (**Vulnerable individual**) peuvent être distingués selon leur catégorie, comme les **Pregnant Woman** ou les **Newborn**. Le concept d'exposition peut utiliser de nombreux éléments de l'Exposure Ontology, notamment **Exposure outcome** et **Exposure stimulus**. Les deux premiers niveaux de la hiérarchie des classes de l'ontologie sont illustrés dans la figure 2, où, par souci de clarté, nous n'affichons pas notre préfixe. Compte tenu de ces concepts, nous pouvons énumérer les plus importantes propriétés de l'ontologie. Un PFAS **has_chemical_structure** composé d'atomes de carbone tandis que **Observation** représente la mesure spécifique d'un PFAS (**measured_pfas**) dans un région géographique (**in_occupation**) et dans un lieu (**is_measured_in**). Une activité **uses** ou **produces** des produits, et un produit **contains** PFAS. Enfin, une instance d'exposition **concerns** PFAS et **affects** des personnes. À ce stade, nous pouvons déjà définir certains axiomes qui nous aideront à classer les instances :

$\texttt{Perfluoroalkyl} \sqcap \texttt{Polyfluoroalkyl} \sqsubseteq \bot$.
$\texttt{Long chain} \sqcap \texttt{Short chain} \sqsubseteq \bot$.
$\texttt{Non polymer} \sqcap \texttt{Polymer} \sqsubseteq \bot$.
$\texttt{Short chain} \equiv \exists \texttt{has\_chemical\_structure}.(\texttt{number\_of\_atoms} \leq 7)$.
$\texttt{Long chain} \equiv \exists \texttt{has\_chemical\_structure}.(\texttt{number\_of\_atoms} > 7)$.

3.2 Evaluation de l'ontologie

L'évaluation d'une ontologie est généralement basée sur le domaine, les utilisateurs et de la spécificité des questions de compétence auxquelles l'ontologie doit répondre. Les requêtes SPARQL suivantes permettent de répondre aux questions de compétence. Par souci de concision et de limitation d'espace, les préfixes ne sont pas affichés.

(i)
```
SELECT ?C ?x ?y WHERE {
?C rdfs:subClassOf :PFAS. ?x a ?C.
?x :has_chemical_structure ?y.}
```

(ii)
```
SELECT DISTINCT ?a WHERE {
?pfas rdf:type :PFAS.
?p rdf:type :Product.
?p :contains ?pfas.
?a :produces ?p.}
```

(iii)
```
SELECT DISTINCT ?e WHERE {
?m rdf:type :Measurement.
?o rdf:type :Occupation.
?m :in_occupation ?o.
?o :in_environment ?e.}
```

(iv)
```
SELECT ?d ?m WHERE {
?d :has_method ?m.}
```

(v)
```
SELECT ?l WHERE {
?m :is_measured_in ?l.
?m a :Observation.
?m :hasSimpleResult ?v.
?m :resultTime ?t.
?m :measured_pfas ?p.
FILTER NOT EXISTS {
?hatm a :Observation.
?hatm :resultTime ?hatt.
?hatm :is_measured_in ?l.
?hatm :measured_pfas ?p.
FILTER (?hatt > ?t). }
FILTER ({ SUM(?v) >= k }) } }
```

(vi)
```
SELECT ?l ?t WHERE {
?m a :Observation.
?m :resultTime ?t.
?m :is_measured_in ?l.}
```

(vii)
```
SELECT ?class (COUNT(?c) WHERE {
?pfas a :PFAS. ?m :measures ?pfas.
FILTER (?class = :Short chain ||
```

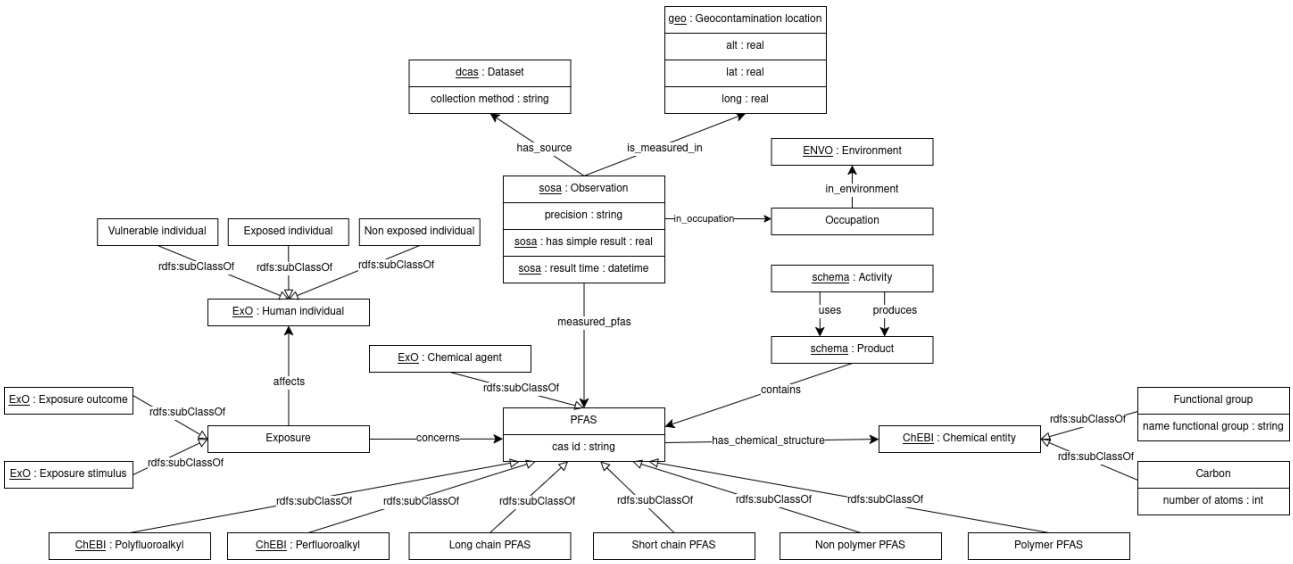

FIGURE 2 – Diagramme pour l'ontologie OntoPFAS

```
?class = :Long chain) }
GROUP BY ?class ORDER BY DESC(?c)
```

La requête (i) extrait toutes les sous-classes de PFAS et, pour chaque instance de la sous-classe, nous récupérons des informations sur leur structure chimique. La requête (ii) extrait toutes les activités générant un produit contenant des PFAS. La requête (iii) identifie tous les types d'environnements dans lesquels les PFAS ont été détectés. La requête (iv) récupère toutes les méthodes permettant d'obtenir des mesures dans les PFAS. La requête (v) récupère tous les lieux où la concentration globale de PFAS dépasse un certain seuil, en veillant à ne conserver que la dernière mesure effectuée pour un même PFAS dans un même lieu. La requête (vi) récupère toutes les mesures de PFAS dans un lieu spécifique, en prenant en compte l'heure des observations. Enfin, la requête (vii) recense les PFAS les plus fréquemment mesurés en fonction de leur catégorie chimique.

Au-delà des réponses aux **questions de compétence**, nous nous appuyons sur des **principes et lignes directrices** modernes afin de garantir une évaluation conforme aux principes généraux des **Linked Data** [8] et du **FAIR** (*Findable, Accessible, Interoperable, Reusable*) [4]. En suivant les recommandations de Villalón et al. [16], nous vérifions la conformité aux principes FAIR :

— **Findable** : attribution d'identifiants aux métadonnées, ajout de métadonnées descriptives (étiquette, version et commentaire) et publication dans **GitHub**.

— **Accessible** : réutilisation de métadonnées existantes et publication de l'ontologie sur GitHub [9], garantissant un accès via un protocole sécurisé avec authentification et autorisation.

— **Interoperable** : utilisation du langage OWL, standard formel, partagé et accessible, ainsi que la réutilisation de ressources existantes pour plusieurs concepts clés.

— **Reusable** : mise à disposition des données sous licence **GNU General Public License v3.0** et des (méta)données via le glossaire standard Dublin Core.

Notre ontologie satisfait 14 des 17 recommandations FAIRsFAIR [9] et est ainsi considérée comme un vocabulaire 5 étoiles [10].

4 Conclusion

Dans cet article, nous avons développé une première ontologie dédiée à la représentation des PFAS dans l'environnement et à leur exposition aux personnes . Nous avons adapté la méthodologie LOT afin d'assurer une modélisation cohérente et exploitable. Cette première version permet dejà de répondre à des questions de compétence pertinentes dans le domaine. Elle servira de base à l'alignement et à l'intégration de nouvelles sources de données, facilitant ainsi l'interopérabilité avec d'autres ontologies et jeux de données existants. Les perspectives d'évolution de ce travail incluent l'élargissement du périmètre à d'autres domaines, l'enrichissement des données sur les PFAS et l'introduction de règles avancées pour le raisonnement, afin d'améliorer l'analyse et la détection d'anomalies environnementales.

Remerciements

Ce travail bénéficie du soutien conjoint de l'Institut EposSUM et de la Mission pour les Initiatives Transverses et Interdisciplinaires (MITI) du CNRS.

8. https://www.w3.org/DesignIssues/LinkedData.html
9. https://github.com/davide1797/PFAS.git
10. https://bvatant.blogspot.com/2012/02/is-your-linked-data-vocabulary-5-star_9588.html

Références

[1] V Richard Benjamins and Dieter Fensel. The ontological engineering initiative (ka) 2. In *Formal Ontology in Information Systems*, pages 287–301. Citeseer, 1998.

[2] Robert C Buck, James Franklin, Urs Berger, Jason M Conder, Ian T Cousins, Pim De Voogt, Allan Astrup Jensen, Kurunthachalam Kannan, Scott A Mabury, and Stefan PJ van Leeuwen. Perfluoroalkyl and polyfluoroalkyl substances in the environment : terminology, classification, and origins. *Integrated environmental assessment and management*, 7(4) :513–541, 2011.

[3] Robert C Buck, Stephen H Korzeniowski, Evan Laganis, and Frank Adamsky. Identification and classification of commercially relevant per-and polyfluoroalkyl substances (pfas). *Integrated environmental assessment and management*, 17(5) :1045–1055, 2021.

[4] Neil P Chue Hong, Daniel S Katz, Michelle Barker, Anna-Lena Lamprecht, Carlos Martinez, Fotis E Psomopoulos, Jen Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, et al. Fair principles for research software (fair4rs principles). *Zenodo*, 2022.

[5] Alissa Cordner, Phil Brown, Ian T. Cousins, Martin Scheringer, Luc Martinon, Gary Dagorn, Raphaëlle Aubert, Leana Hosea, Rachel Salvidge, Catharina Felke, Nadja Tausche, Daniel Drepper, Gianluca Liva, Ana Tudela, Antonio Delgado, Derrick Salvatore, Sarah Pilz, and Stéphane Horel. Pfas contamination in europe : Generating knowledge and mapping known and likely contamination with "expert-reviewed" journalism. *Environmental Science & Technology*, 58(15) :6616–6627, 2024.

[6] Nikolai Dahlem and Axel Hahn. User-friendly ontology creation methodologies-a survey. *AMCIS 2009 Proceedings*, page 117, 2009.

[7] Nina De Lille and Ben Roelens. A practical application of upon lite for the development of a semi-informal application ontology. In *15th International Workshop on Value Modelling and Business Ontologies*, pages 63–70. CEUR-WS, 2021.

[8] Paula de Matos, Adriano Dekker, Marcus Ennis, Janna Hastings, Kenneth Haug, Steve Turner, and Christoph Steinbeck. Chebi : a chemistry ontology and database. *Journal of cheminformatics*, 2 :1–1, 2010.

[9] Ingrid Dillo. General overview of fairsfair. 2021.

[10] Mariano Fernández-López, Asuncion Gomez-Perez, and Natalia Juristo. Methontology : from ontological art towards ontological engineering. *Engineering Workshop on Ontological Engineering (AAAI97)*, 03 1997.

[11] Fatma-Zohra Hannou Hannou, Victor Charpenay, Maxime Lefrançois, Catherine Roussey, Antoine Zimmermann, and Fabien Gandon. La méthodologie acimov pour l'intégration agile et continue des modules ontologiques. In *35èmes Journées francophones d'Ingénierie des Connaissances (IC 2024)-Plateforme d'intelligence artificielle (PFIA 2024)*, pages 127–128, 2024.

[12] Zhibo Lu, Rong Lu, Hongyuan Zheng, Jing Yan, Luning Song, Juan Wang, Haizhen Yang, and Minghong Cai. Risk exposure assessment of per-and polyfluoroalkyl substances (pfass) in drinking water and atmosphere in central eastern china. *Environmental Science and Pollution Research*, 25 :9311–9320, 2018.

[13] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101 : A guide to creating your first ontology, 2001.

[14] Silvio Peroni. Samod : an agile methodology for the development of ontologies. In *Proceedings of the 13th OWL : Experiences and Directions Workshop and 5th OWL reasoner evaluation workshop (OWLED-ORE 2016)*, pages 1–14, 2016.

[15] Helena Sofia Pinto, Steffen Staab, and Christoph Tempich. Diligent : Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In *ECAI*, volume 16, page 393, 2004.

[16] María Poveda-Villalón, Paola Espinoza-Arias, Daniel Garijo, and Oscar Corcho. Coming to terms with fair ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 255–270. Springer, 2020.

[17] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. Lot : An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111 :104755, 2022.

[18] Abdul Sattar, Ely Salwana Mat Surin, Mohammad Nazir Ahmad, Mazida Ahmad, and Ahmad Kamil Mahmood. Comparative analysis of methodologies for domain ontology development : A systematic review. *International Journal of Advanced Computer Science and Applications*, 11(5), 2020.

[19] Thiago G Schwanz, Marta Llorca, Marinella Farré, and Damià Barceló. Perfluoroalkyl substances assessment in drinking waters from brazil, france and spain. *Science of the total environment*, 539 :143–152, 2016.

[20] Barry Smith. Ontology. In *The furniture of the world*, pages 47–68. Brill, 2012.

[21] Daniele Spoladore, Elena Pessot, and Alberto Trombetta. A novel agile ontology engineering methodology for supporting organizations in collaborative ontology development. *Computers in Industry*, 151 :103979, 2023.

Vers une approche basée sur les graphes de connaissances pour l'évaluation de la qualité des données dans l'IoT

Oumaima Amal[1], Nathalie Hernandez[2], Thierry Monteil[1]

[1] IRIT, Université de Toulouse, INSA Toulouse, France
[2] IRIT, Université Toulouse 2 Jean Jaurès, Toulouse, France

prénom.nom@irit.fr

Résumé

L'évaluation de la qualité des données (QD) dans les systèmes IoT constitue un défi en raison de l'hétérogénéité des données, de leur production en flux continus et des environnements dynamiques dans lesquels elles sont produites. Les approches existantes se limitent souvent à un nombre restreint d'indicateurs de qualité. Cet article propose une approche granulaire et traçable reposant sur des graphes de connaissances (GC) pour enrichir et annoter les données avec des informations détaillées sur leur qualité. Celle-ci prend en compte divers indicateurs en fonction du contexte de production des données et de la tâche cible. Trois niveaux de granularité sont définis afin de structurer l'évaluation des indicateurs de QD, et des ontologies dédiées à cette évaluation sont présentées dans le cadre de ce travail.

Mots-clés

QD, granularités d'évaluation, GC, ontologies, énergie

Abstract

Evaluating data quality (DQ) in IoT systems is challenging due to their heterogeneity, real-time generation, and the dynamic environments in which they operate. Existing approaches often focus on a limited set of quality indicators. This paper proposes a granular and traceable approach based on knowledge graphs (KG) to enrich and annotate data with detailed quality information. It considers various indicators depending on the data production context and the target task. Three levels of granularity are defined to structure the evaluation of DQ indicators, and ontologies dedicated to this assessment are presented as part of this work.

Keywords

DQ, assessement granularities, KG, ontologies, energy

1 Introduction

L'évaluation de la qualité des données (QD) constitue une étape incontournable dans la gestion des données. Face à la quantité massive de données générées, reçues ou stockées par les dispositifs interconnectés, leur transmission requiert une réflexion approfondie sur leur qualité afin de garantir une optimisation des données transmises. L'étude [13] propose un cadre parmi les plus couramment utilisés, structurant un ensemble d'indicateurs de QD en quatre catégories : intrinsèque, contextuelle, représentationnelle et accessibilité. Certaines analyses, comme celle de [12], distinguent les indicateurs s'appliquant à une valeur unique de ceux s'appliquant à un ensemble de données.

Dans ce travail, nous proposons des niveaux de granularité spécifiques aux environnements IoT. En effet, l'évaluation de la QD ne se limite pas à étudier une ou plusieurs valeurs, mais à considérer la ou les valeurs dans le contexte dans lequel elles sont produites. Nous nommons une valeur et son contexte une observation. L'évaluation d'un ensemble de données diffère selon que ces observations proviennent d'un même dispositif ou de plusieurs dispositifs distincts. L'objectif de cet article est alors d'introduire une approche permettant de calculer des indicateurs de QD en prenant en compte ces niveaux de granularité. Une valeur ajoutée de cette approche réside dans l'utilisation d'ontologies, assurant une traçabilité du processus d'évaluation de la QD.

Cet article est structuré comme suit : la Section 2 introduit les indicateurs de QD et décrit les ontologies utilisées pour représenter les observations, notamment *Sensor, Observation, Sample, and Actuator (SSN/SOSA)*[1] et *Extensible Observation Ontology (OBOE)*[2], ainsi que celle dédiée à la modélisation de la QD dont *Data Quality Vocabulary (DQV)*[3]. La Section 3 introduit les différents niveaux de granularité d'évaluation, une revue des indicateurs selon ces niveaux et décrit l'ontologie proposée *Data Quality Assessment (DQA)*. La Section 4 présente les principales étapes de notre approche pour l'évaluation de la QD.

2 État de l'art : Indicateurs de QD et Ontologies

Dans cet article, nous considérons un exemple de base simple pour l'étude des indicateurs de QD et des ontologies permettant de les décrire. Nous prenons comme cas d'usage illustratif cinq dispositifs installés dans une salle de classe d'un bâtiment d'un campus. Nous considérons cinq capteurs, dont deux sont des capteurs de présence, et trois

1. https://www.w3.org/TR/vocab-ssn/
2. https://bioportal.bioontology.org/ontologies/OBOE
3. https://www.w3.org/TR/vocab-dqv/

sont intégrés à trois plateformes : un climatiseur, un vidéo-projecteur et un système de ventilation. Ces trois capteurs sont utilisés pour suivre leur consommation énergétique (en *watts*) sous forme d'observations. Ce cas d'usage est décrit via des ontologies qui seront introduites dans les sections suivantes du papier (Voir figure 2).

2.1 Indicateurs de QD existants

Les indicateurs de QD ont été largement étudiés dans la littérature. L'un des cadres les plus couramment utilisés pour les classifier est celui proposé par [13], qui définit quatre grandes catégories de QD : intrinsèque, contextuelle, représentationnelle et accessibilité (Voir figure 1). Nous ne considérons pas l'étude des indicateurs liés à l'accessibilité qui pourra être réalisée dans des travaux futurs, notamment dans une perspective d'intégration de mécanismes de sécurité et de contrôle d'accès. Nous introduisons les définitions des indicateurs sans prendre en considération les méthodes utilisables pour les quantifier, notre objectif est de fournir un cadre général se concentrant sur leur finalité.

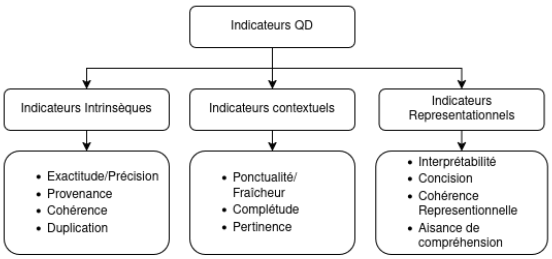

FIGURE 1 – Catégories et indicateurs de la QD

Indicateurs intrinsèques : cette catégorie inclut des dimensions qui peuvent être évaluées indépendamment de la tâche pour laquelle les données produites seront utilisées. L'évaluation de la QD intrinsèque se concentre sur les propriétés inhérentes à la valeur des données elle-même. Nous présentons les définitions des indicateurs intrinsèques :

Précision : Dans [9], les auteurs définissent la précision comme le degré avec lequel les valeurs mesurées sont proches les unes des autres. Nous retenons cette définition.

Exactitude : Dans [4] et [1], les auteurs décrivent l'exactitude comme le degré de proximité d'une valeur par rapport à une valeur de référence dans le domaine, tandis que pour [5], l'exactitude est appelée correction et correspond au degré avec lequel les données sont conformes à une valeur attendue ou à un ensemble de valeurs acceptables prédéfinies. Pour nous, l'exactitude est le degré avec lequel la valeur est proche d'une valeur réaliste attendue. e.g. : le capteur `sensor#08` rapporte à `2025-01-31T09:00:00` une observation (`obs_020`) avec une valeur de `330.00W`. La consommation attendue pour la plateforme `videoprojector#65` est entre `100W` et `500W`. Comme la valeur observée est dans l'intervalle, l'exactitude est fixée à `100%`.

Provenance : Dans [10], la provenance est la possibilité de connaître la source des données ainsi que la description de cette source. Elle fait référence au suivi et à la documentation de l'origine des données et du processus par lequel les données ont été générées, collectées ou transformées. Nous utiliserons la même définition dans ce travail. e.g. : l'observation `obs_003` est générée par le capteur `sensor#53`, qui est hébergé sur la plateforme `airconditionner#32`.

Cohérence : Selon [6], [9], et [11], la cohérence des données est définie comme le degré, exprimé en pourcentage, avec lequel deux ou plusieurs valeurs dans un ensemble de données ne sont pas contradictoires. Nous suivons cette définition en considérant que ces valeurs, qu'elles proviennent d'un seul appareil ou de plusieurs sources produisant des données similaires, doivent respecter des règles ou des contraintes prédéfinies, telles que des relations logiques ou des lois physiques. e.g. l'observation `obs_015`, capturée par le capteur `sensor#08` à `2025-01-31T09:00:00`, indique une puissance consommée de `290W` pour `videoprojector#65`. Cependant, si l'observation suivante `obs_016`, réalisée par le même capteur, mesure `600W`, cette variation soudaine et inexpliquée diminue le degré de cohérence des observations.

Duplication : Selon [4], la duplication fait référence à la présence de valeurs distinctes pour le même attribut de la même entité dans un ensemble de données. Nous considérons alors qu'il y a une duplication lorsque deux valeurs identiques ou plus sont présentes dans un ensemble de données, bien qu'elles soient censées être distinctes en fonction de la fréquence attendue de génération des observations. e.g. Deux observations distinctes, `obs_004` et `obs_003`, générées par le capteur `sensor#53` à la même date et heure (`2025-01-31T08:30:00`), contiennent exactement la même valeur de consommation énergétique (`2200.0W`). Cela indique une duplication des données.

Indicateurs contextuels : cette catégorie englobe des attributs qui dépendent du contexte d'utilisation, en particulier en lien avec la tâche à accomplir. Contrairement à la QD intrinsèque, la QD contextuelle ne peut être évaluée sans prendre en compte le contexte ou/et l'objectif pour lesquels les données sont utilisées. Nous introduisons les définitions des indicateurs contextuels :

Ponctualité : Selon [9], la ponctualité fait référence à l'âge des données, qui doit être approprié pour la tâche à accomplir. Elle évalue si les données sont disponibles au moment requis et respectent les délais prévus pour leur utilisation. Nous suivons cette définition en considérant que des données ponctuelles doivent être reçues et traitées dans un délai compatible avec les exigences de la tâche cible. e.g. supposons qu'une tâche nécessite des observations dans les 30 dernières minutes en temps réel. Si le capteur `sensor#53` fournit des données toutes les `15 min` et que l'observation `obs_010` arrive avec un retard de `20 min`, elle dépasse la fenêtre de `30 min` et devient obsolète, ce qui peut impacter les résultats obtenus.

Complétude : Selon [11], la complétude correspond au degré avec lequel les données fournissent les valeurs at-

tendues pour les caractéristiques requises dans un contexte d'utilisation donné. Dans [12], elle est définie par la présence de valeurs pour toutes ces caractéristiques. Pour nous, elle correspond à la présence de valeurs ou instances attendues pour les propriétés et entités requises afin de répondre aux besoins d'une tâche spécifique. e.g. les observations `obs_001` à `obs_010` sont considérées complètes si et seulement si le capteur capture des valeurs pour la propriété observée `power consumption` pour toutes ces observations.

Pertinence : Pour [7], la pertinence fait référence aux données répondant aux besoins pour lesquels elles ont été collectées, placées dans une base de données et utilisées. Pour nous, la pertinence fait référence au degré d'importance des valeurs des données pour un objectif prédéfini. Elle est relative et dépend forcément de l'évaluation des autres indicateurs de qualité. e.g. les observations `obs_001` à `obs_010` sont pertinentes si et seulement si elles sont complètes.

Indicateurs représentationnels : cette catégorie se concentre sur le format et l'interprétabilité des données. Elle inclut les indicateurs suivants :

Interprétabilité : La description de la donnée attendue est claire en termes de langue, symbole et unité. De plus, la donnée considérée peut être associée à la description correspondante.

Concision : La concision est définie par [13] et [2] comme le degré avec lequel les données sont représentées de manière compacte.

Cohérence Représentationnelle : Dans [9], il est mentionné que la cohérence représentationnelle fait référence au fait que les données doivent être présentées dans des formats compatibles.

Facilité de Compréhension : Selon [5], la compréhension est liée à la clarté des données, qui doivent être sans ambiguïté. Pour nous, l'évaluation de ce critère repose sur l'analyse des trois indicateurs précédents.

2.2 SOSA et OBOE : Ontologies pour la description des observations

Selon [8], l'ontologie *SOSA* [4] est un cadre léger conçu pour modéliser et décrire les interactions et les fonctionnalités des capteurs, des actionneurs, ainsi que les processus d'observation et d'échantillonnage. Elle a été développée dans le cadre d'un effort collaboratif entre l'Open Geospatial Consortium (OGC) et le World Wide Web Consortium (W3C) afin de répondre aux besoins des développeurs web, des scientifiques de domaine et des ingénieurs en données liées. Dans le cadre de l'exemple considéré au début de cet article (Voir la figure 2), la salle de classe est modélisée comme une `sosa:FeatureOfInterest`, identifiant le contexte spatial des observations représentées par la classe `sosa:Observation`.

4. https://github.com/w3c/sdw

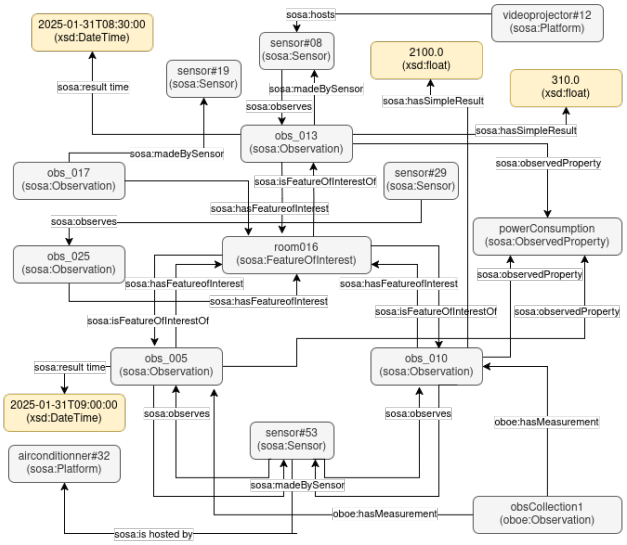

FIGURE 2 – Exemple illustratif de GC avec SOSA et OBOE

Les observations, générées par des capteurs définis à travers la propriété `sosa:madeBySensor`, mesurent la consommation d'énergie via la propriété `powerConsumption` de type `sosa:observedProperty`, indiquant *ce qui* est mesuré. La donnée produite par un capteur est exprimée via la propriété `hasSimpleResult` lorsqu'il s'agit d'une valeur numérique simple. Enfin, l'instant de production de chaque observation est enregistré grâce à la propriété `sosa:resultTime`, permettant de la situer dans son contexte temporel. Chaque observation, comme `obs_003` ou `obs_020`, suit ce même schéma de description. Au-delà de la description fournie par SOSA, l'ontologie *OBOE* facilite la représentation d'une collection d'observations `oboe:Observation` qui peut être liée à plusieurs observations `sosa:Observation` (l'équivalent de `oboe:Measurement` [5]) via la propriété `oboe:hasMeasurement`.

2.3 DQV : Ontologie pour l'évaluation de la QD

L'ontologie *Data Quality Vocabulary - DQV* [6] (Voir la figure 4), constitue un cadre général pour décrire la QD. Bien qu'il ne vise pas à fournir une définition formelle et exhaustive de la qualité, il établit une approche cohérente pour partager des informations de qualité, permettant aux utilisateurs d'évaluer la pertinence d'un jeu de données pour leurs besoins spécifiques. Conçue comme une extension du vocabulaire *Data Catalog Vocabulary - DCAT* [7], DQV s'appuie sur les travaux antérieurs de représentation de la QD, notamment l'ontologie *Dataset Quality Vocabulary - daQ* [3], qui se concentre sur la capture d'informations en particulier des métriques, relatives à la qualité des jeux de données. DQV va au-delà en proposant une structure plus flexible, intégrant des aspects tels que les annotations, les politiques

5. https://www.w3.org/ns/sosa/oboe
6. https://www.w3.org/TR/vocab-dqv/
7. https://www.w3.org/TR/vocab-dcat-3/

et les certifications de qualité, offrant ainsi une description plus riche.

DQV comprend des classes principales telles que `dqv:Dimension`, qui représente les dimensions ou indicateurs de la qualité. Il y a aussi `dqv:Category`, qui permet de classer ces dimensions en différentes catégories. La classe `dqv:Metric` définit les métriques associées à chaque dimension. Enfin, `dqv:QualityMeasurement` associe les valeurs des mesures obtenues en appliquant ces métriques. Les propriétés principales incluent `dqv:inDimension` reliant une métrique à la dimension correspondante, `dqv:inCategory` reliant une dimension à une catégorie, `dqv:expectedDataType` qui spécifie le type de données attendu pour chaque métrique, et `dqv:value` qui attribue une valeur mesurée à une métrique. En complément, des propriétés issues de *SKOS* (*Simple Knowledge Organization System*), comme `skos:prefLabel` et `skos:definition`, enrichissent la représentation des dimensions et des métriques, en précisant leurs labels et définitions.

3 Granularités d'évaluation de QD et Ontologie DQA

3.1 Niveaux de granularité

L'étude [12] est l'une des rares à explorer la notion de granularité dans l'évaluation de la QD, en faisant la distinction entre points de données individuels (cellules) et ensembles de données. Bien qu'intéressante, elle ne prend pas en compte les cas intermédiaires ni les défis posés par les données IoT, où les observations peuvent être liées par le temps, l'espace ou le contexte. Nous proposons une catégorisation qui évalue la QD à travers trois niveaux (Voir figure 3), permettant ainsi de mieux structurer l'évaluation des indicateurs de QD dans l'approche qui est prévue pour être déployable dans un réseau d'objets IoT :

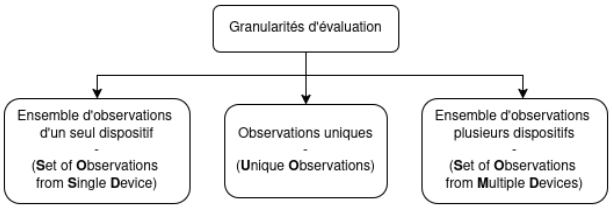

FIGURE 3 – Granularités d'évaluation

- **Observations uniques, UO :** Celles-ci représentent des observations individuelles sur une mesure unique à un instant précis.
- **Ensemble d'observations provenant d'un seul dispositif, SO-SD :** À ce niveau, les observations proviennent d'un seul dispositif et mesurent une même propriété sur un intervalle de temps.
- **Ensemble d'observations provenant de plusieurs dispositifs, SO-MD :** Les observations, à ce niveau, sont issues de plusieurs dispositifs portant sur une ou plusieurs propriétés qui sont considérées en fonction de l'objectif d'utilisation de ces observations.

3.2 Indicateurs revus par rapport aux niveaux

Dans cette partie, nous présentons une analyse des indicateurs introduits dans 2.1 en fonction des niveaux de granularités proposés dans 3.1, comme le montre le tableau 1. Les indicateurs marqués par * sont évalués selon des conditions supplémentaires au niveau SO-MD.

Indicateur	UO	SO-SD	SO-MD
Exactitude	X		
Précision		X	X*
Provenance	X	X	X
Cohérence		X	X
Duplication		X	X*
Complétude		X	X*
Ponctualité		X	X
Pertinence	X	X	X
I. Représentationels	X	X	X

TABLE 1 – Indicateurs par niveaux de granularité

Le croisement des indicateurs de QD avec les niveaux de granularité permet de mettre en évidence que l'évaluation de ces indicateurs peut varier en fonction du niveau de granularité. L'application de tous les indicateurs nécessite l'acquisition des valeurs des observations (e.g. les valeurs associées aux observations `obs_001`, `obs_002`, ...), les capteurs qui les génèrent (e.g. `sensor#53`, `sensor#19`, ...), les plateformes qui les hébergent (e.g. `airconditionner#32`, `ventilationsys#07`, ...) et qui sont en fonctionnement ainsi que la propriété observée (e.g. `powerConsumption`). Certains indicateurs sont évalués uniquement au niveau UO, ce qui permet de garantir un marquage de qualité unitaire pour chaque valeur observée ; c'est le cas de l'*exactitude* par exemple. Par ailleurs, au niveau de SO-SD qui peuvent être constituées de plusieurs observations UO, l'évaluation des indicateurs de QD est requise en raison de son importance pour faciliter la sélection des observations ayant un niveau de qualité suffisant pour être soit comparées, soit fusionnées avec d'autres observations pour des évaluations ultérieures de QD au niveau SO-MD. Pour les ensembles d'observations SO-SD et SO-MD qui ne sont pas marqués par (*), les indicateurs de QD sont évalués selon les définitions présentées dans la section 2.1. Cependant, certains indicateurs, tels que la *complétude*, la *précision* et la *duplication*, peuvent nécessiter des conditions particulières d'application, notamment au niveau SO-MD. Ces conditions sont expliquées via l'exemple suivant. Pour clarifier, prenons l'exemple de deux ensembles d'observations, `obsCollection1` et `obsCollection2`, générées respectivement par les capteurs `sensor#53` et `sensor#73`, qui mesurent la même propriété `powerConsumption` et sont hébergés par des plateformes du même type. Pour la complétude, si les deux capteurs générant `obsCollection1`

et `obsCollection2` ont respectivement des pas de temps différents (e.g. `5min` et `10min`), il devient pertinent d'évaluer si les deux ensembles peuvent être combinés pour former une collection unique. Cela permettrait, par exemple, de compléter les données manquantes dans une collection. Concernant l'évaluation de la précision, il est également possible de combiner les deux ensembles `obsCollection1` et `obsCollection2` pour évaluer la précision globale d'une seule mesure (e.g. `powerConsumption`). Enfin, pour l'évaluation de la duplication, il est possible de comparer les ensembles `obsCollection3` et `obsCollection4` issus des capteurs de présence `sensor#19` et `sensor#20`, à condition qu'ils observent le même phénomène d'intérêt, mesurent la même propriété aux mêmes timestamps.

3.3 DQA : Extension proposée pour DQV

Data Quality Assessment - DQA[8] est une extension proposée de l'ontologie *DQV* (Voir la figure 4). Elle enrichit le modèle *DQV* en introduisant de nouveaux concepts et relations pour une évaluation plus détaillée et traçable de la QD. *DQA* conserve les classes fondamentales de *DQV*, notamment `dqv:Dimension`, `dqv:QualityMeasurement`, et `dqv:Metric`.

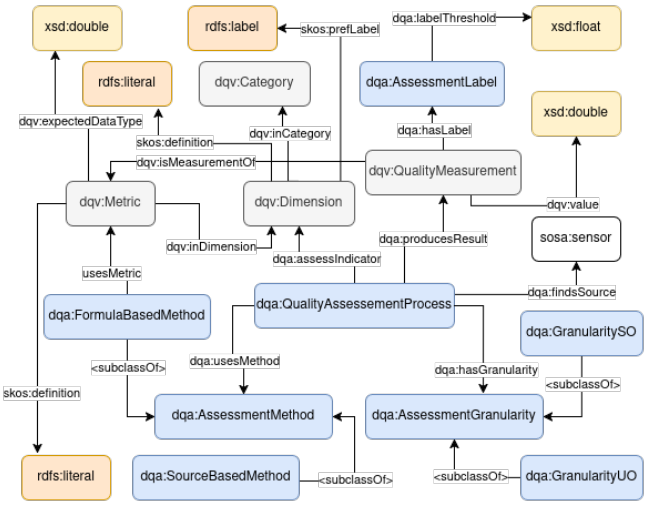

FIGURE 4 – Vocabulaires DQA/DQV

DQA introduit `dqa:QualityAssessmentProcess`, comme une classe qui permet de décrire le processus d'évaluation de la QD pour garantir la traçabilité des méthodes utilisées. Ces méthodes sont définies par la classe `dqa:AssessmentMethod`. Les méthodes basées sur des formules sont représentées par `dqa:FormulaBasedMethod`, tandis que celles permettant de récupérer les sources et leurs descriptions contenues dans le GC sont décrites par `dqa:SourceBasedMethod`. La propriété `dqa:findsSource` est associée aux méthodes `dqa:SourceBasedMethod` et permet de décrire la source retrouvée dans le GC, si elle existe. (Voir exemples[9] pour la provenance, l'exactitude et la duplication.)
Afin de structurer davantage l'évaluation, *DQA* introduit la classe `dqa:AssessmentGranularity`, qui précise le niveau de granularité des évaluations. Cette classe se décline en `dqa:GranularitySO` pour un *Set of Observations* et `dqa:GranularityUO` pour une *Unique Observation*. De plus, la classe `dqa:AssessmentLabel` permet d'étiqueter les résultats des évaluations et est associée à un seuil, défini par la propriété `dqa:labelThreshold`. Par ailleurs, la propriété `dqa:assessIndicator` relie un processus d'évaluation à un indicateur de qualité, tandis que `dqa:producesResult` capture la sortie d'un processus d'évaluation sous forme de mesure de qualité. L'association entre un processus et une méthode spécifique est représentée par `dqa:usesMethod`, et la granularité de l'évaluation est précisée par `dqa:hasGranularity`. Enfin, la mesure obtenue est qualifiée par un label à l'aide de la propriété `dqa:hasLabel`.

4 Vers une approche d'évaluation granulaire et traçable de la QD

L'approche suit trois étapes principales, comme le montre la figure 5, et s'appuie sur les ontologies SOSA, OBOE, DQV et DQA pour assurer la modélisation des processus d'évaluation des indicateurs de QD tout en prenant en compte les niveaux de granularité. Des exemples pour chaque étape sont introduits sur GitLab[10].

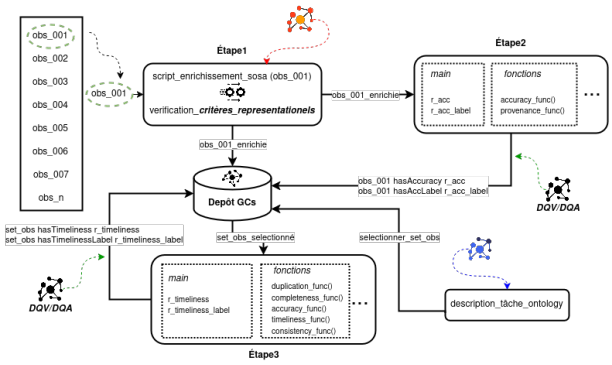

FIGURE 5 – Aperçu sur les trois étapes de l'approche

Étape 1 : Enrichissement de l'observation
La première étape consiste à recevoir une observation et à l'enrichir en utilisant l'ontologie *SOSA*, qui permet de structurer les relations entre les capteurs, les observations et les propriétés observées. Contrairement aux étapes suivantes, cette phase ne nécessite pas une description des traitements effectués sur l'observation en termes d'indicateurs représentationnels, parce qu'ils concernent principalement la bonne représentation des données plutôt que leur évaluation qualitative. Une fois l'enrichissement terminé, l'observation est stockée dans le dépôt de GC

8. https://gitlab.irit.fr/projet-ai-nrgy/dqa-ontology.git
9. https://gitlab.irit.fr/projet-ai-nrgy/dqa-ontology/provenance
10. https://gitlab.irit.fr/projet-ai-nrgy/dqa-ontology/examples

pour être utilisée dans les phases d'évaluation.

Étape 2 : Évaluation des indicateurs QD au niveau UO
À cette étape, les indicateurs de QD pouvant être évalués sur une observation unique (*UO*) sont appliqués. Ceci inclut : l'*exactitude*, la *provenance*, la *pertinence* et la *fraîcheur*. Chaque indicateur est évalué individuellement, et le résultat obtenu est stocké dans le dépôt avec une description complète du processus d'évaluation en utilisant les ontologies *SOSA*, *DQV* et *DQA*.

Étape 3 : Évaluation des indicateurs QD aux niveaux SO-SD et SO-MD
Dans cette dernière étape, l'évaluation porte sur des ensembles d'observations (*SO-SD* et *SO-MD*). Deux types d'évaluation peuvent être réalisés :

- **Évaluation agnostique de la tâche** : certains indicateurs de QD peuvent être évalués indépendamment d'un objectif utilisateur, uniquement en tenant compte du contexte de génération des observations. Ces indicateurs incluent : la *précision*, la *duplication* et la *cohérence*.
- **Évaluation liée à une tâche définie par l'utilisateur** : lorsqu'un utilisateur définit une tâche spécifique, d'autres indicateurs de QD peuvent être nécessaires en plus des trois indicateurs précédents, dont la *complétude*, la *ponctualité* et la *pertinence*.

L'ontologie *OBOE* est utilisée pour définir les ensembles d'observations sur lesquels l'évaluation est effectuée. Les résultats de l'évaluation sont documentés via *DQV*, le processus d'évaluation étant décrit en utilisant *DQA*.

5 Conclusion

Cet article propose une approche basée sur les GC pour l'évaluation de la QD dans les systèmes IoT, avec une contribution principale : la structuration des indicateurs QD selon différents niveaux de granularité (*UO*, *SO-SD*, *SO-MD*). Dans ce travail, nous nous sommes concentrés sur le cas des dispositifs fixes, vu que l'intégration des dispositifs mobiles augmentera la complexité d'application des indicateurs de QD, mais pourra être creusée dans nos futurs travaux. Cette organisation granulaire permet une évaluation plus fine et adaptée au contexte de production des données. Nous avons également proposé une extension de *DQV*, nommée *DQA*, et réutilisé les ontologies *SOSA*, *OBOE* et *DQV* pour décrire et sauvegarder les métadonnées associées au processus de l'évaluation des indicateurs de QD. Nous avons défini une méthodologie pour mettre en œuvre cette approche. Le travail à venir consistera à finaliser l'implémentation en se basant notamment sur SPARQL et des règles SHACL distribuées. Cette approche sera validée dans un contexte énergétique dans le cadre du projet *AI-NRGY*.

Remerciements

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre de France 2030 portant la référence 22-PETA-0004 - projet AI-NRGY.

Références

[1] Carlo Batini and Monica Scannapieca. *Data Quality : Concepts, Methodologies and Techniques*. Springer, Berlin, 2006.

[2] John Byabazaire, Gregory O'Hare, and Declan Delaney. Data quality and trust : Review of challenges and opportunities for data sharing in iot. *Electronics, by MDPI*, 20(24) :1–20, Dec 2020.

[3] Jeremy Debattista, Christoph Lange, and Sören Auer. daq, an ontology for dataset quality information. In *Proceedings of the Linked Data on the Web Workshop (LDOW 2014), by CEUR-WS.org*, Seoul, Korea, 2014.

[4] Christopher Fox, Anany Levitin, and Thomas Redman. The notion of data and its quality dimensions. *Inf Process Manag, by Elsevier*, 30(1) :9–19, 1994.

[5] Christian Fürber and Martin Hepp. Using semantic web resources for data quality management. In *Knowledge Engineering and Management by the Masses, by Springer*, pages 211–225, Berlin, Heidelberg, 2010.

[6] Sandra Geisler, Christoph Quix, Sven Weber, and Matthias Jarke. Ontology-based data quality management for data streams. *Journal of Data and Information Quality, by ACM*, 7(4) :1–34, 2016.

[7] Thomas Nelson Herzog, Fritz Joseph Scheuren, and William Edward Winkler. *Data Quality and Record Linkage Techniques*. Springer, New York, 2007.

[8] Krzysztof Janowicz, Armin Haller, Simon J.D. Cox, Danh Le Phuoc, and Maxime Lefrançois. Sosa : A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics, by Elsevier*, 56 :1–10, 2019.

[9] Nisrine Makhoul. Review of data quality indicators and metrics, and suggestions for indicators and metrics for structural health monitoring. *Advances in Bridge Engineering, by Springer Nature*, 3(1) :1–32, 2022.

[10] Maria Priestley, Fionntán O'Donnell, and Elena Simperl. A survey of data quality requirements that matter in ml development pipelines. *Journal of Data and Information Quality, by ACM*, 15(2) :3592616, 2023.

[11] R. Pérez-Castillo, A. G. Carretero, I. Caballero, M. Rodríguez, M. Piattini, A. Mate, S. Kim, and D. Lee. Daqua-mass : An iso 8000-61 based data quality management methodology for sensor data. *Sensors, by MDPI*, 18(9) :3105, 2018.

[12] Antonio Vetrò, Lorenzo Canova, Marco Torchiano, Camilo Orozco Minotas, Raimondo Iemma, and Federico Morando. Open data quality measurement framework : Definition and application to open government data. *Government Information Quarterly, by Elsevier Ltd*, 33(2) :325–337, 2016.

[13] Richard Y. Wang and Diane M. Strong. Beyond accuracy : What data quality means to data consumers. *Journal of Management Information Systems, by Taylor Francis*, 12(4) :5–33, 1996.

Session 4 : Modélisation et vérification formelle dans des contextes industriels

Kalamar : un langage de modélisation à base de règles

Pauline Armary[1,2], Fabien Givors[1], Antoine Spicher[1], Sattisvar Tandabany[1]

[1] Anabasis, 93200 Saint-Denis, France
[2] CIAD UR 7533, Université Bourgogne Europe, F-21000 Dijon, France

pauline.armary@ube.fr,[givors,spicher,tandabany]@anabasis-assets.com

Résumé

Kalamar est un langage de modélisation à base de règle issu des exigences d'un contexte industriel de modélisation. Ce contexte requiert efficacité et précision lors de la modélisation et lisibilité lors de sa restitution aux clients. Avec Kalamar, nous proposons une syntaxe de haut niveau pour la modélisation par règle.

Mots-clés

Ontologie, langage formel, modélisation, règle

Abstract

Kalamar is a rule-based modeling language created for and from an industrial practice. This industrial context requiers precision and efficiency in the modeling process and readability for the client. We present, with Kalamar, a high-level syntax for rule-based modeling.

Keywords

Ontology, formal language, modelisation, rule

1 Introduction

Le langage Kalamar est né d'un besoin de modélisation dans un contexte industriel de réponse à des enjeux complexes de traitement de l'information. Il est le résultat des améliorations successives initiées en réponse aux exigences des projets entrepris : interopérabilité entre différents systèmes d'information, vérification de la bonne application de contraintes réglementaires au sein d'un logiciel, gestion des droits d'accès aux données et à des informations stratégiques en fonction de différents profils d'utilisateurs, reconstruction du parcours de soin ou de réparation d'un appareil industriel à partir de plusieurs sources de données hétérogènes [1].

La démarche entreprise dans toutes ces expériences repose sur le rapprochement de la modélisation et du développement logiciel exécutant cette modélisation [16], au point que le modèle soit l'application, et que toute modification du modèle corresponde exactement à une modification de l'application. Ce rapprochement s'est concrétisé par la création du langage Kalamar répondant aux enjeux miroir :

Modélisation par les ingénieurs : le langage doit offrir une grande efficacité et de la précision dans la modélisation, donnant les clefs aux ingénieurs pour définir les éléments précis d'information au plus proche des exigences des clients, en effaçant autant que possible les contraintes liées à l'expressivité du langage.

Restitution aux clients : le langage doit permettre au client de comprendre ce qui a été implémenté, pour garantir une maîtrise a posteriori de l'ensemble des informations : avoir un regard métier directement sur les informations traitées et pouvoir modifier certains éléments en cas de changement spécifique (par exemple, un changement de réglementation).

Parmi les techniques de modélisation possibles, le parti pris a été de s'inscrire dans le cadre du web sémantique tel que standardisé par le W3C [1], et en particulier de la modélisation ontologique. L'objectif est de réaliser une modélisation du domaine métier en faisant abstraction de l'existant technologique et des contraintes auxquelles il doit répondre. La compréhension du domaine métier est pensée comme le centre de la pratique de modélisation, dont la connexion avec son expression dans un système d'information n'est que la partie nécessaire à son implémentation. Nous appelons *ingénieur de la connaissance* la personne en charge de la modélisation ontologique de ces connaissances métier.

Parmi les différentes propositions qui ont émergé au sein du web sémantique, le choix a été pris de s'inspirer des approches par règles, en particulier de l'utilisation des règles existentielles portée par RuleML [6], ainsi que par les langages logiques tels que Datalog [14] ou Prolog [9]. Ce choix est la conséquence d'un double constat : (i) les approches par règles (en format « *si* . . . *alors* . . . ») sont mieux comprises par les clients que des présentations en termes de hiérarchies de classes, en particulier lorsqu'elles incluent des expressions de classes complexes (comme pour OWL [15]) ; (ii) ces approches facilitent également la traçabilité des connaissances car tout fait inféré peut être justifié par la cascade de règles appliquées lors de sa déduction, apportant aux clients une explication dans des termes qu'il a lui-même compris et validés.

Si différentes propositions existent pour la spécification de règles, celles-ci proposent des syntaxes souvent minimalistes et/ou se concentrant sur des problématiques d'échange. Ce type de considération n'est pas propre aux standards de spécification de règles ; on citera par exemple la syntaxe Manchester [10] pour OWL. Le besoin de proposer un langage à base de règles de plus haut niveau est déjà

1. https://www.w3.org/

identifié [7]. On pourra citer les langages SWRL [11] et Notation3 [5, 2]. Nous nous inscrivons dans la même perspective, tout en cherchant une syntaxe aisée dans son utilisation par les ingénieurs de la connaissance et lisible pour les clients qui doivent s'assurer de la conformité des règles avec leur métier.
Nous présentons dans cet article les principales caractéristiques du langage de modélisation par règle Kalamar. La section 2 présente les exigences d'une entreprise de modélisation et les besoins en termes de langage à base de règles. La section 3 présente la syntaxe du langage Kalamar et la sémantique associée. La section 4 illustre son utilisation sur un exemple de modélisation simple. Enfin, l'article se conclut en section 5 par une discussion sur la proposition Kalamar et les perspectives attendues.

2 Contexte et motivations

Le langage Kalamar est né des activités industrielles de modélisation à base d'ontologie. Les besoins en termes de langage de modélisation sont issus de ces expériences. Nous commençons par décrire les exigences d'une entreprise de modélisation, pour ensuite identifier à travers un exemple simple de modélisation les primitives que le langage de règles doit proposer.

2.1 Exigences de modélisation

L'ensemble des expériences industrielles de modélisation et l'étude de l'existant dans le cadre du web sémantique, nous ont amené à considérer les exigences suivantes pour une syntaxe étendue.

Une syntaxe de haut niveau : Proposer les abstractions nécessaires afin de représenter des concepts métier complexes, facilitant la manipulation des modèles par les experts du domaine et réduisant l'écart cognitif entre le formalisme technique et la réalité opérationnelle.

Une syntaxe à base de règles : Utiliser comme construction fondamentale le format simple et compréhensible des règles si-alors et pouvoir profiter des outils existants (implémentations des langages Prolog et Datalog, moteurs d'inférence tels que Jena [8], RDFox [13], Graal [4], etc.).

Une compatibilité avec les standards existants : Reposer sur les standards du web sémantique pour assurer la compatibilité et l'interopérabilité des modèles avec les outils et les méthodes existants, capitaliser sur l'expérience des utilisateurs déjà familiarisés avec ces conventions pour finalement tenter de remporter leur adhésion à la syntaxe proposée.

Une expressivité étendue mais maîtrisée : Proposer un outil réalisant de façon efficace les abstractions apportées par le langage et permettant le développement de modèles exécutables et traçables tout en garantissant l'absence d'incohérences par la spécification de contraintes appropriées.

Ces exigences ont amené à la création du langage de modélisation à base de règles Kalamar.

2.2 Modélisation par règles

Pour mieux comprendre les besoins spécifiques en modélisation par règles qui ont été rencontrés lors de nos différentes expériences industrielles, nous proposons d'analyser un exemple jouet mais inspiré d'un projet réel. Le thème abordé est *le suivi du parcours vaccinal d'un patient dans le cadre de la crise COVID*. En voici une description informelle simplifiée.

> *L'objectif est de déterminer le statut vaccinal d'un patient en fonction des règles épidémiologiques en vigueur. Un patient est une personne particulière prise en charge et dont on connaît les vaccins reçus et les résultats de dépistage. Son statut vaccinal est dit* complet *lorsque le patient présente 2 vaccinations, ou une vaccination et un test positif à la maladie. Les vaccinations et les tests COVID sont des évènements particuliers. Les évènements sont comparables en fonction de leur date. Si le statut vaccinal du patient n'est pas complet, un rappel doit lui être envoyé trois mois après la dernière vaccination pour prévoir une deuxième vaccination. Les vaccinations sont reconnues grâce à des certificats de vaccination. Le système doit assurer que les vaccinations stockées sont bien associées à des certificats.*

Il est possible de définir un modèle de cette situation comme un ensemble de règles logiques si-alors, comme l'aurait implémenté un programmeur Prolog par exemple. Cependant, on retrouve dans ce type de modélisation logique des motifs de règles qui sont sans cesse réutilisés. Le langage Kalamar propose de réifier ces motifs en primitives de haut niveau construites sur la simple notion de règle. Bien que concis, l'exemple du suivi vaccinal rend compte de cet ensemble de fonctionnalités. En effet, pour modéliser cette situation, l'ingénieur de la connaissance doit définir les éléments suivants.

Classes et propriétés : le parcours de vaccination est centré autour du patient et des vaccinations qui lui sont associées. Les ingénieurs de la connaissance ont besoin de définir rapidement et aisément une classe et les propriétés qui les mettent en rapport.

Héritage : le patient est une personne particulière pour laquelle des informations de santé sont connues en plus des données personnelles usuelles. Les ingénieurs de la connaissance doivent pouvoir isoler les information de santé des données personnelles, pour des questions de confidentialité et pour permettre une extension du modèle (comme la modélisation des médecins).

Règle : « si un patient a deux preuves de vaccination, alors il a un statut vaccinal complet. » Les ingénieurs de la connaissance ont besoin d'identifier deux objets associés à un même patient, pour en déduire une nouvelle propriété attachée à celui-ci.

Test : la vaccination est un objet numérique qui représente un évènement réel. Pour permettre de confirmer sa bonne exécution dans le monde réel, la vaccination

est attachée à un certificat de vaccination, qui sert de preuve de réalisation de la vaccination. Si un évènement de vaccination existe mais qu'aucun certificat n'est associé, une alerte sur l'incomplétude de la base de donnée doit être remontée.

Comparaison entre données typées : pour permettre de vérifier la séquence de vaccination, il est nécessaire de déterminer si un date est antérieure à une autre. Les ingénieurs de la connaissance ont ainsi besoin d'effectuer des comparaisons entre des données typées, notamment des dates.

Calcul sur les données typées : sachant qu'une période de 3 mois doit être respectée entre chaque vaccination, les médecins veulent pouvoir définir la date à partir de laquelle une prochaine vaccination est possible. Les ingénieurs de la connaissance doivent avoir la capacité d'effectuer des calculs sur des données typées, notamment sur les dates.

Négation : les médecins veulent également connaître les patients qui n'ont jamais été vaccinés. Les ingénieurs de la connaissance ont ainsi besoin de pouvoir exprimer une forme de négation dans la modélisation.

Agrégateur : les médecins veulent connaître la date de dernière vaccination d'un patient. Celle-ci est déterminée en identifiant la vaccination dont la date est la plus grande. Les ingénieurs de la connaissance veulent ainsi pouvoir effectuer des agrégations sur des propriétés associées à une classe, pour capturer le plus grand élément parmi l'ensemble.

Existentielle : pour informer le patient, les médecins souhaitent qu'un rappel soit envoyé à la date de prochaine vaccination, pour que celui-ci planifie son prochain rendez-vous de vaccination. Les ingénieurs doivent ainsi avoir la possibilité de créer des nouveaux éléments dans la modélisation en fonction de règles définies.

Le langage Kalamar propose une réponse à ces différents besoins. Nous le détaillons dans la section suivante.

3 Le langage Kalamar

Afin de répondre aux contraintes de développement industriel (notamment l'utilisation d'outils tiers et la nécessité de flexibilité pour des évolutions futures), le langage Kalamar a été conçu comme l'extension d'un moteur de raisonnement (abstrait) à base de règles avec des éléments syntaxiques de haut-niveau. Nous présentons dans cette section la sémantique de l'inférence attendue, suivie de la présentation du langage sur la base de cette inférence.

3.1 Un moteur de raisonnement minimaliste

L'une des motivations initiales de Kalamar était l'extension d'un moteur capable de réaliser une inférence simple. L'implémentation de ce moteur est laissée à des outils tiers dont nous supposons qu'ils proposent un jeu minimal de fonctionnalités. Nous introduisons ces fonctionnalités que nous présentons formellement. Les exigences minimales répertoriées sont les suivantes :

Reposer sur le standard RDF : le moteur infère un graphe de connaissances RDF, c'est-à-dire un ensemble de triplets dont chaque élément est un URI. Nous supposons donc connus l'ensemble du vocabulaire et des recommandations RDF/RDFS. En particulier, nous supposons un ensemble de constantes littérales (standard XSD par exemple) pour manipuler les valeurs simples telles que les entiers, les chaînes de caractères, etc. Enfin, le développement du langage Kalamar ne fait pas appel aux nœuds anonymes de RDF (*blank nodes*).

Utiliser des clauses de Horn : dans la veine du langage Datalog, nous supposons que le moteur de raisonnement est capable de manipuler des règles simples de la forme $a_1 \wedge a_2 \wedge \cdots \Rightarrow a$ permettant d'inférer l'atome a lorsque la conjonction d'atomes $a_1 \wedge a_2 \wedge \ldots$ est vraie.

Permettre des prédicats *built-in* : nous supposons enfin que le moteur est capable d'accepter la création de *built-in*, c'est-à-dire de permettre d'enrichir les triplets de bases de prédicats avec l'ajout de fonctionnalités calculatoires utilitaires. On pense notamment aux *built-in* du langage de règle SWRL. C'est également à travers ce mécanisme d'enrichissement que Kalamar étend ce cœur minimal avec des constructions telles que les existentielles ou les négations par l'échec. Ces prédicats étant prédéfinis à l'inférence et sans effet de bord (Kalamar est purement déclaratif), nous considérons que l'invocation des *built-in* est interdite en tête de règle.

Ces exigences ont amené à la formalisation suivante. Elle repose sur les ensembles I des URI, V des constantes littérales, B des *built-in* et X des variables. Une fonction

$$|.| : B \to \mathbb{N} \cup \{\infty\}$$

appelée *arité* précise le nombre d'arguments attendus par un *built-in*. L'arité ∞ sera utilisée pour des *built-in* avec un nombre arbitraire d'arguments. Les *atomes* respectent la grammaire

$$a ::= \langle e, e, e\rangle \mid b(e, \ldots, e)$$

où $e \in I \cup V \cup X$ et $b \in B$.
Les *triplets* s'écrivent $\langle s, p, o\rangle$ avec le *sujet* s, le *prédicat* p et l'*objet* o; s et p sont restreints à $I \cup X$. Les appels aux *built-in* s'écrivent $b(a_1, \ldots, a_n)$ et doivent respecter l'arité de b lorsque $|b| \neq \infty$. Nous appellerons *faits*, les triplets et les appels de *built-in* sans variable.
Les *règles* sont de la forme

$$a_1 \wedge \ldots \wedge a_n \Rightarrow \langle s, p, o\rangle$$

avec $n \in \mathbb{N}$. La partie gauche d'une règle est appelée le *corps* et la partie droite la *tête*. Une règle est bien formée lorsque les variables apparaissant dans sa tête apparaissent au moins une fois dans son corps. Il faut comprendre une règle comme universellement quantifiée sur les variables

de son corps. Lorsque le corps est vide, nous écrirons simplement $\top \Rightarrow \langle s, p, o\rangle$ dénotant l'inférence d'un fait sans hypothèse. On dénote $\mathcal{R}$ l'ensemble de toutes les règles.
Une *substitution* est une fonction partielle de $X \rightharpoonup V$, notée σ. On étend les substitutions à $X \cup V$ par $\sigma(\alpha) = \alpha$ pour toute constante $\alpha \in V$. L'application d'une substitution sur un atome a s'écrit $\sigma[a]$ et est définie par

$$\sigma[\langle s, p, o\rangle] := \langle \sigma(s), \sigma(p), \sigma(o)\rangle$$
$$\sigma[b(a_1, \ldots, a_n)] := b(\sigma(a_1), \ldots, \sigma(a_n)).$$

Étant donnée une conjonction d'atomes $c := a_1 \wedge \cdots \wedge a_n$, on écrit $\mathrm{Subst}(c)$ l'ensemble des substitutions applicables à c.
L'ensemble de ces définitions nous permet d'aboutir à la formalisation de l'inférence. Étant donné un ensemble de faits K, un ensemble de règles R et un fait f, on écrira $K, R \vdash f$ pour exprimer que *le fait f est déductible de l'ensemble de faits K et de l'ensemble des règles R*. Cette relation est définie par

$$\frac{f \in K}{K, R \vdash f}$$

$$\frac{\bigwedge_i a_i \Rightarrow a \in R \quad \sigma \in \mathrm{Subst}(\bigwedge_i a_i) \quad K, R \vdash \sigma[a_i]}{K, R \vdash \sigma[a]}.$$

Nous retrouvons dans cette construction les éléments attendus de la modélisation ontologique : la ABox (*assertion box*) sous la forme de K et la TBox (*terminology box*) sous la forme de R.

3.2 La syntaxe Kalamar

Le langage Kalamar se présente comme un ensemble de constructions syntaxiques étendant les règles précédentes. Ainsi, un programme Kalamar P est transformé en un ensemble de règles que nous écrirons $[\![P]\!]$. La sémantique du programme P sera alors donnée par $-, [\![P]\!] \vdash -$ associant à un graphe de connaissances K l'ensemble des faits inférés. Dans la suite de la section, nous introduisons la syntaxe des programmes P sous la forme d'une grammaire BNF étendue à mesure que nous présentons les constructions du langage. L'effet de $[\![-]\!]$ sera alors décrit informellement.
Nous commençons par présenter le noyau de la syntaxe de Kalamar permettant la spécification de règles.

```
<program> ::= <rule>*

<rule> ::= rule <uri> <anonymous_rule>

<anonymous_rule> ::=
  "{" if <body> then <head> "}"

<body> ::=
    <body> (and | or) <body>
  | "(" <body> ")"
  | <body_atom>

<head> ::=
  ( (<var_uri> <var_uri> <var_uri> |
    <anonymous_rule>) and? )*

<body_atom> ::=
    true
  | <var_uri> <var_uri> <var_uri>
  | <uri> "(" (<var_uri> ",")* ")"

<var_uri> ::= <var> | <uri>
```

Les symboles terminaux `<uri>` et `<var>` correspondent respectivement à $I \cup V \cup B$ et X. Les notations infixes pour les *built-in* correspondant aux opérations mathématiques binaires et aux comparateurs, ont été omises.

Un programme Kalamar se présente sous la forme d'une collection de règles. Celles-ci présentent trois caractéristiques supplémentaires par rapport aux règles de $\mathcal{R}$ du moteur. Chacune d'elles peut être dépliée en un ensemble de règles classiques.

Règles avec plusieurs têtes. Une première facilité syntaxique consiste à autoriser l'inférence d'une conjonction de triplets en tête de règle. Le dépliement consiste alors à démultiplier la règle, une pour chaque tête. La règle

```
if ?x pèreDe ?y
then
  ?x parentDe ?y and ?y enfantDe ?x
```

se développe en deux règles.

```
if ?x pèreDe ?y then ?x parentDe ?y
if ?x pèreDe ?y then ?y enfantDe ?x
```

Disjonctions d'hypothèses. Kalamar permet également d'éviter des duplications de code en autorisant les disjonctions dans les corps des règles. Le dépliement consiste à nouveau à démultiplier la règle pour chaque cas. Il est alors demandé que les ensembles des variables de chaque alternative soient les mêmes. Ainsi, la règle

```
if ?x filleDe ?y or ?x filsDe ?y
then ?x enfantDe ?y
```

se développe en deux règles.

```
if ?x filleDe ?y then ?x enfantDe ?y
if ?x filsDe ?y then ?x enfantDe ?y
```

D'autres facilités d'écriture concernant les prédicats (omises dans le présent article) sont également proposées. Par exemple, la disjonction précédente peut s'écrire plus simplement `?x (filleDe | filsDe) ?y`.

Règles imbriquées. Enfin, Kalamar permet l'insertion d'une règle (anonyme) dans la tête d'une autre règle. Cela permet de factoriser des règles dont seule une partie du corps est commune. On peut par exemple écrire

```
if ?x parentDe ?y
then
  ?y enfantDe ?x and
  { if ?y parentDe ?z
    then ?x grandParentDe ?z }
```

qui se déplie en deux règles.

```
if ?x parentDe ?y then ?y enfantDe ?x
```

```
if ?x parentDe ?y and ?y parentDe ?z
then ?x grandParentDe ?z
```

3.3 Les schémas en Kalamar

Nous introduisons ici les éléments relatifs à l'expression des schémas dans le respect du standard RDF/RDFS. Nous abordons ainsi les notions usuelles de classe et de propriété, ainsi que leurs traductions sous forme de règles.
Un schéma spécifie essentiellement des contraintes de typage. En RDF, les nœuds d'un graphe de connaissances peuvent être typés grâce au prédicat particulier `rdf:type`. Cette construction est tellement importante en modélisation, que le choix a été d'intégrer au langage l'idiome classique `a` pour y référer. La syntaxe des `<body_atom>` et des `<head_atom>` est alors étendue avec la construction `<var_uri> "a" <var_uri>`. Les types auxquels font références ces triplets dans leur composante objet sont issus de la spécification du schéma. La syntaxe Kalamar des schémas est la suivante :

```
<program> ::=
  (<rule> | <property> | <class>)*

<property> ::=
  property <uri> <subproperty_of>? ":"
    <uri> "->" <uri> <property_defs>?

<property_defs> ::=
  "{" (is <property_def> ","?)* "}"

<property_def> ::=
    reflexive
  | transitive
  | symmetrical
  | inverseOf <uri>

<class> ::=
  class <uri> <subclass_of>? "{"
    <class_property>* "}"

<class_property> ::=
   has <uri> ":" <uri> <property_defs>?

<subproperty_of> ::=
   subPropertyOf (<uri> ","?)*

<subclass_of> ::=
   subClassOf (<uri> ","?)*
```

Cette syntaxe permet de définir des propriétés et des classes.

Les propriétés. On entend par *propriétés* les prédicats déclarés de type `rdf:Property`. Elles peuvent spécifier leur domaine (`rdfs:domain`) et leur co-domaine (`rdfs:range`). On remarquera la possibilité de spécifier des propriétés classiques comme la réflexivité, la symétrie, etc. Les règles correspondantes seront alors générées par l'opérateur $[\![-]\!]$. Par exemple, on pourra spécifier la propriété `parentDe` des exemples précédents par :

```
property parentDe:
  Personne -> Personne { is transitive }
```

Les règles suivantes seront dérivées de ce programme.

```
if true then
  parentDe a rdf:Property and
  parentDe rdfs:domaine Personne and
  parentDe rdfs:range Personne

if ?x parentDe ?y and ?y parentDe ?z
then ?x parentDe ?z
```

La notion de sous-propriété (`rdfs:subPropertyOf`) est également présente. On pourra ainsi spécifier la propriété `pèreDe`.

```
property pèreDe subPropertyOf parentDe
```

correspondant aux règles suivantes.

```
if true then
  pèreDe a rdf:Property and
  pèreDe rdfs:subPropertyOf parentDe

if ?x pèreDe ?y then ?x parentDe ?y
```

Les classes. On entend par *classes* le type RDF `rdfs:Class` des objets du prédicat `rdf:type`. Les classes sont également utilisées lors de la spécification des signatures de propriétés. Le sous-typage (`rdfs:subClassOf`) est également autorisé. La partie de la syntaxe Kalamar dédiée aux classes permet de regrouper l'ensemble des contraintes d'une classe dans une seule déclaration. Ainsi la spécification Kalamar

```
class Personne {
  has nom: xsd:string
  has naissance: xsd:date
}

class Femme subClassOf Personne
```

se développe en les règles et propriétés suivantes :

```
if true then
  Personne a rdfs:Class and
  Femme a rdfs:Class and
  Femme rdfs:subClassOf Personne

if ?x a Femme then ?x a Personne

property nom: Personne -> xsd:string
property naissance: Personne -> xsd:date
```

3.4 Existentielles

L'une des fonctionnalités essentielles identifiées lors d'expériences industrielles est la capacité à créer de nouveaux nœuds dans la base de connaissances. Pour cela, Kalamar fournit un mécanisme d'existentielles. Il faut distinguer ce mécanisme du quantificateur existentiel de la logique, où ce dernier est souvent associé aux nœuds anonymes RDF [12] avec l'idée d'implémenter l'opérateur à travers la skolémisation des formules. En Kalamar, on entend par existentielle la création d'un nouveau nœud *nommé* de façon unique à partir d'un ensemble d'informations connues. Étant donné qu'une base de triplets ne peut contenir de nœuds isolés, le nœud nouvellement créé doit apparaître dans un triplet. Il a

été décidé de créer uniquement des nœuds typés. L'extension de la BNF suivante présente les éléments syntaxiques correspondant à ce mécanisme.

```
<program> ::= ( ... |  <function> )*

<function> ::=
  fun <uri> "(" (<var> ",")* ")"
    ":" <uri> "{" <head>? "}"

<head> ::=
  ( ( ... | <existential> ) and? )*

<existential> ::=
  exists <uri> "(" (<var_uri> ",")* ")"
    (as <var>)?
```

Cette syntaxe introduit tout d'abord les *fonctions* permettant de spécifier la construction de nouveaux objets. Une fonction est caractérisée par son nom, ses arguments, le type des objets construits, et un ensemble d'atomes de têtes. Par exemple,

```
fun enfant(?m, ?p, ?_ordre): Personne {
  ?m mèreDe $ and
  ?p pèreDe $
}
```

définit la fonction `enfant` construisant de nouveaux objets de classe `Personne` étant donnés trois objets (arguments `?m`, `?p` et `?_ordre`) correspondant aux parents de la personne nouvellement créée ainsi que l'ordre de naissance dans sa fratrie. On remarquera l'utilisation de `$`, une variable spéciale de l'ensemble V dénotant l'objet construit. On remarquera aussi que `?_ordre` n'est utilisée ici que pour rendre unique l'URI de chaque enfant mais n'intervient pas dans le corps de la fonction. L'invocation au constructeur ne peut se faire qu'en tête de règles à l'aide du mot-clé `exists` comme illustré ci-dessous.

```
if ... then
  exists enfant(?x, ?y, ?num) as ?z and ...
```

On suppose que les variables `?x`, `?y` et `?num` sont définies par ailleurs. La variable `?z` quant à elle permet de référer à l'objet nouvellement créé et peut être utilisée dans l'ensemble de la tête. À noter que si la variable n'est utilisée nulle part ailleurs dans la tête, alors le `as ?z` peut être omis.

Afin d'exprimer la sémantique de cette nouvelle construction, nous considérons un *built-in* spécial $\exists \in B$ et variadique ($|\exists| = \infty$) dont l'appel prend la forme suivante

$$\exists(x, f, v_1, \ldots, v_n)$$

où f est l'URI de la fonction, v_i celui du i^{e} argument passé à f, et x celui de l'objet construit. Le rôle de ce *built-in* (dont l'implémentation est laissée de coté dans cet article) est simplement de calculer l'URI unique x en fonction des autres arguments. Son invocation est transférée dans le corps de la règle, alors que le contenu de la fonction est développée dans la tête. Ainsi, l'invocation à la fonction `enfant` précédente est dépliée comme suit.

```
if ... and ∃(?z, enfant, ?x, ?y, ?num) then
  ?z a Personne and
  ?x mèreDe ?z and
  ?y pèreDe ?z and
  ...
```

3.5 Agrégations et NAF

Certaines règles nécessitent d'évaluer leur condition sur un groupe d'individus, pour les compter par exemple. Ce genre de constructions est bien établi dans le domaine des bases de données relationnelles avec le `GROUP BY` de SQL et les opérations d'agrégations (`COUNT`, `SUM`, `MAX`, etc.). L'idée est de porter ces constructions dans le langage Kalamar. Pour se faire, la syntaxe du corps des règles est augmentée de la façon suivante :

```
<body_atom> ::= ...
  | <var> "=" <aggregator> "{"
      <var> where <body> "}"

<aggregator> ::=
  count | sum | max | min
```

Ce type de construction peut bien sûr mener à des paradoxes. Une façon classique de prévenir de tels écueils consiste à étager un ensemble de règles R en une famille d'ensembles disjoints R_i tels que $R = \bigcup_i R_i$ et où les agrégations de l'ensemble R_i ne peuvent faire référence qu'à des atomes inférables par des règles des ensembles R_j avec $j < i$. On supposera donc que les règles spécifiées autorisent un tel étagement. Dans ce cas, la sémantique d'une agrégation au niveau i est basée sur les faits inférés aux niveaux inférieurs. Nous définissons alors la sémantique des agrégations relativement un ensemble de faits K qu'il faut comprendre comme les faits inférés dans les niveaux inférieurs. À chaque agrégateur $\alpha \in \{\texttt{count}, \texttt{sum}, \ldots\}$, on associe une famille de *built-in* $\alpha_{K,x,\varphi}$ d'arité 1, où K est un ensemble de faits, φ une conjonction d'atomes et x une variable de φ. La sémantique est donnée directement par

$$\alpha_{K,x,\bigwedge_i a_i}(n) \;:\Longleftrightarrow\; \alpha\{\sigma(x) \mid \sigma \in \mathrm{Subst}(\textstyle\bigwedge_{\mathrm{i}} \mathrm{a_i}) \wedge \forall i.\sigma[a_i] \in K\}.$$

Un atome de type agrégation

```
?n = α { ?x where φ }
```

sera simplement traduit en l'appel *built-in* $\alpha_{K,?\mathtt{x},\varphi}(?\mathtt{n})$ où K dénote l'ensemble des faits inférés aux niveaux inférieurs.

Négation par l'échec. Il est possible de dériver des agrégateurs une forme de négation consistant à vérifier qu'une requête ne retourne aucune résultat. Ce type de négation est appelée *négation par l'échec* (en anglais NAF pour *Negation as Failure*) et est disponible en Kalamar.

```
<body_atom> ::= ...
  | naf "{" <body> "}"
```

Nous n'en détaillerons pas la sémantique qui est similaire à celle d'un agrégateur de type `count`.

3.6 Tests & contraintes d'intégrité

Les constructions présentées jusqu'ici avaient pour rôle d'inférer de nouvelles connaissances. Cependant, afin de vérifier l'intégrité d'une modélisation ou des données manipulées, il est souvent nécessaire de considérer des règles test spécifiant un ensemble de contraintes attendues. Le langage Kalamar propose un tel dispositif à travers les éléments syntaxiques suivants.

```
<program> ::= ( ... | <test> )*

<test> ::=
  test <uri> "{"
    when <body>
    assert <body>
    or else ( warn | error ) <string>
  "}"
```

Un test est une forme de règle particulière constituée de deux ensembles d'atomes (homogènes à des corps de règles classiques) spécifiant que la réalisation du premier doit entraîner la réalisation du second. Le sens donné à une règle test `when` φ `assert` ψ `or else warn` θ est celui d'une règle classique de la forme `if` φ `and naf` $\{\psi\}$ `then exists Warning(...) as ?w and ?w message` θ où la tête de la règle spécifie la génération d'un fait d'erreur. Nous omettons délibérément cette partie qui repose sur une ontologie des tests Kalamar dont la présentation sort du périmètre de cet article.

3.7 Réalisation

Une implémentation propriétaire[2] du langage Kalamar a été développée. Elle est bâtie au-dessus du *triple store* et du raisonneur d'Apache/Jena [8] qui réalise l'inférence $\vdash$. Le plus gros projet réalisé, Cortex [1], comportait plus d'une centaine de règles Kalamar (soit environ un millier de règles après compilation $[\![-]\!]$) et raisonnait sur un graphe de plusieurs centaines de millions de nœuds (individus). La complexité de l'inférence repose sur les capacités du moteur sous-jacent pouvant utiliser des techniques connues telles que l'algorithme de RETE.

4 Exemple de modèle Kalamar

Pour concrétiser l'utilisation du langage Kalamar, nous reprenons l'exemple présenté dans la section 2, et décrivons la manière dont celui-ci peut être modélisé dans la syntaxe Kalamar.

4.1 Définir les classes et propriétés

La modélisation commence avec la construction de l'objet principal du processus, le patient, qui hérite des informations personnelles d'une classe `Personne`. Les propriétés associées au patient, telle que la vaccination reçue et le statut vaccinal sont également définies à l'intérieur de la classe. L'ingénieur de la connaissance décrit ainsi une classe à laquelle sont attachées des propriétés typées et les relations à d'autres objets qui sont centrales à cet objet. La syntaxe Kalamar se présente ainsi :

```
class Personne {
  has nom: xsd:string
  ...
}

class Patient subClassOf Personne {
  has recoitVaccination: Vaccination
  has statutVaccinal: StatutVaccinal
  has aPreuveVaccination: CertifVaccination
  has nombre_vaccination: xsd:int
}
```

Les accolades associées permettent de définir les informations associées à la classe via le mot-clef `has`, sur le modèle des cadres (en anglais *frames*) proposés par la syntaxe Manchester [10]. Cette méthode de définition permet de grouper l'ensemble des informations associées à une même entité au sein d'une même syntaxe, pour accroître la lisibilité et mieux retrouver l'information au moment de l'édition. Habituellement, les propriétés sont associées à la classe qui correspond à leur domaine, qui est donc implicite. Le co-domaine de la propriété est exprimé après les `:`, comme dans l'exemple `has recoitVaccination: Vaccination`.
Les propriétés peuvent également être définies en dehors de ce cadre de la classe, en définissant le domaine et le co-domaine de manière indépendante, lorsque la lisibilité l'impose.

```
class CertifVaccination

class CertifVaccination1
  subClassOf CertifVaccination

class CertifVaccination2
  subClassOf CertifVaccination
```

La classe `CertifVaccination` englobe tous les certificats de vaccination, à savoir les certificats de première vaccination, sous-typés par la classe `CertifVaccination1` et les certificats de deuxième vaccination, sous-typés par la classe `CertifVaccination2`.
On modélise le concept de certificat de vaccinations par une super classe afin que la propriété `aPreuveVaccination` puisse avoir comme co-domaine n'importe quel sous-type de `CertifVaccination`
De la même manière on modélise les événements de vaccination. On modélisera plus précisément la classe `Event` en section 4.5.

```
class Vaccination subClassOf Event

class PremiereVaccination
  subClassOf Vaccination

class SecondeVaccination
  subClassOf Vaccination
```

2. Documentation : https://karnyx.com/doc/en/.

4.2 Déterminer le statut vaccinal du patient

La première règle concerne la modélisation du statut vaccinal en fonction des vaccinations reçues par le patient. Comme l'objectif est de capturer la vaccination réellement réalisée, la règle s'applique à partir de la preuve de vaccination. La règle se lit ainsi : « Si un patient a un certificat de première vaccination et un certificat de deuxième vaccination, alors il a un statut vaccinal complet ». On notera ici que `statut_complet` est une constante définie par ailleurs, de type `StatutVaccinal`

```
rule statut_vaccinal_complet {
  if
    ?p a Patient and
    ?p aPreuveVaccination ?c1 and
    ?c1 a CertifVaccination1 and
    ?p aPreuveVaccination ?c2 and
    ?c2 a CertifVaccination2
  then
    ?p statutVaccinal statut_complet
}
```

La syntaxe de Kalamar suit en partie l'ordre des mots utilisés dans formulation de la règle en langage naturel, à l'exception de l'introduction des variables (précédées par `?`).

4.3 Patients non vaccinées

Afin de remonter au mieux l'information des patients en attente de vaccination, l'ingénieur de la connaissance souhaite déterminer les patients qui n'ont reçu aucune dose de vaccin. La règle se lit : « Si un patient n'a reçu aucune preuve de vaccination, alors le patient a le statut vaccinal "Incomplet" ».

```
rule total_vaccination {
  if
    ?p a Patient and
    naf { ?p aPreuveVaccination ?c }
  then
    ?p statutVaccinal statut_incomplet
}
```

L'ingénieur de la connaissance utilise le mot-clé `naf` pour vérifier l'absence de preuve de vaccination. La négation par l'échec signifie que le patient est identifié comme non vacciné si le système n'a pas connaissance de sa preuve de vaccination.

4.4 Test d'intégrité des données

Pour valider la conformité entre l'évènement de vaccination déclaré et la réalité effectuée, l'ingénieur de la connaissance définit un test. Il peut se lire ainsi : « Si le patient a reçu une vaccination, vérifier qu'il existe en preuve le certificat de vaccination correspondant, sinon renvoyer un message d'erreur ».

```
test checking_certificat_1 {
  when
    ?p a Patient and
    ?p nom ?nom and
    ?p recoitVaccination ?vacc and
    ?vacc a PremiereVaccination
  assert
    ?p aPreuveVaccination ?certif and
    ?certif a CertifVaccination1
  or else warn
    """Le patient {?nom} a reçu {?vacc}
    mais n'a pas de certificat
    de première vaccination."""@fr
}
```

4.5 Les évènements et l'ordre temporel

Afin de permettre le suivi précis des informations sur les vaccinations, en particulier sur l'ordre des vaccinations et leurs rappels, il est nécessaire de définir une classe évènement à laquelle est associée l'ordre temporel « antérieur à ». Cette propriété est transitive et inverse à la propriété « postérieur à ».

```
class Evenement {
  has date: xsd:date
  has anterieurA: Evenement {
    is transitive,
    inverseOf posterieurA
  }
}
```

Afin de simplifier la modélisation des propriétés, certaines règles sont capturées à travers la syntaxe, en particulier concernant les axiomes des propriétés. Ainsi, la propriété « antérieur à » est définie comme étant transitive, ce qui est retranscrit comme : « Si un évènement A "est antérieur à" un évènement B et que B "est antérieur à" C, alors A "est antérieur à" C ». De même pour le mot-clé `inverseOf` correspond au renversement entre domaine et co-domaine : « Si A "est antérieur à" B, alors B "est postérieur à" A ».

4.6 L'antériorité entre évènements

Pour instancier cette relation d'antériorité entre les évènements, la modélisation s'appuie sur la vérification de l'ordre entre les dates des évènements : « Si un évènement a une date inférieure à un second évènement, alors le premier évènement est antérieur au second ».

```
rule evenement_anterieur {
  if
    ?e1 a Evenement and
    ?e2 a Evenement and
    ?e1.date < ?e2.date
  then
    ?e1 anterieurA ?e2
}
```

Pour plus de facilité, Kalamar introduit les symboles mathématiques classiques de comparaison <, >, = et != sous leur forme infixe. Ils correspondent aux éléments de comparaison tel que définis par SWRL[3]. Le choix a été fait de

3. https://www.w3.org/submissions/SWRL/#8

remplacer la syntaxe extensive de SWRL par les symboles mathématiques, pour des raisons de concision et de lisibilité.

4.7 Date de dernière vaccination

Avec l'ordre des évènements, l'ingénieur de la connaissance peut définir la dernière date de vaccination associée à un patient. La règle se lit : « Si le patient a reçu au moins une vaccination, alors la date de dernière vaccination du patient est la date de la vaccination la plus récente ».

```
rule derniere_vaccination {
  if
    ?p a Patient and
    ?p recoitVaccination ?unVaccin and
    ?derniere_date = max { ?date where
      ?p recoitVaccination ?_v and
      ?_v date ?date
    }
  then
    ?p dernieVaccination ?derniere_date
}
```

L'ingénieur de la connaissance utilise l'agrégation pour identifier la date la plus grande parmi toutes les vaccinations. L'agrégation s'effectue grâce à l'utilisation de l'agrégateur `max`, qui identifie l'élément maximal sur l'ensemble des éléments (`?date`) répondant aux critères précisés après le mot-clé `where`.

4.8 Prochaine date de vaccination

Finalement, l'ingénieur de la connaissance peut calculer la date à partir de laquelle un prochain rappel est possible, en laissant un intervalle d'au moins 3 mois entre les deux vaccinations. La règle se lit : « Si un patient a un statut vaccinal incomplet, alors sa prochaine date de vaccination est définie 3 mois après la date de dernière vaccination. »

```
rule prochaine_date_vaccination {
  if
    ?p a Patient and
    ?p statutVaccinal statut_incomplet and
    ?p dernieVaccination ?dernière and
    swrlb:addYearMonthDurationToDate(
      ?rappel,
      ?dernière,
      "P3M"^^xsd:duration)
  then
    ?p prochaineVaccination ?rappel
}
```

Dans cet exemple, l'ingénieur de la connaissance utilise un *built-in* issu de SWRL pour calculer sur les dates, dans la mesure où cette fonctionnalité n'est pas traduite dans la syntaxe propre de Kalamar. L'ingénieur de la connaissance peut ainsi profiter de fonctionnalités étendues avec les *built-in* SWRL intégrés dans le langage.

4.9 Créer le message de rappel

En dernière étape, l'ingénieur de la connaissance veut créer un message associé au patient à la date de rappel de vaccination, pour l'informer et l'inviter à programmer un rendez-vous de vaccination. La règle se lit : « Si le patient a un statut vaccinal incomplet et qu'il a une date de prochaine vaccination, alors il existe un message de rappel envoyé au patient à la date de prochaine vaccination. »

```
fun msgDeRappelA(?patient, ?rappel)
    : MessageDeRappel {
  $ envoyeA ?patient and
  $ dateDeRappel ?rappel
}

rule rappel_vaccination {
  if
    ?p a Patient and
    ?p statutVaccinal statut_incomplet and
    ?p prochaineVaccination ?rappel
  then
     exists msgDeRappelA(?p, ?rappel)
}
```

La fonction `msgDeRappelA` est un constructeur de message à partir d'une personne et d'une date : `?patient` correspondant à la personne à laquelle le message est envoyée, et `?rappel` à la date d'envoi du rappel. Cette fonction crée un individu de la classe `MessageDeRappel`, définie par ailleurs[4].

La fonction `msgDeRappelA` est appelée dans la tête de la règle `rappel_vaccination` afin de créer l'individu à partir des informations issues du corps de la règle, à savoir dans l'exemple, le `Patient` auquel le message de rappel est envoyé, ainsi que la `dateDeRappel`.

5 Discussion et conclusion

Le langage Kalamar se présente comme une proposition originale de syntaxe de modélisation par règles. Les deux travaux dont il s'inspire sont d'une part les langages de programmation logique, Datalog [14] et Prolog [9], et d'autre part les standards de modélisation ontologiques portés par le W3C, principalement OWL-SWRL et Notation3.

En terme d'expressivité, Kalamar se positionne dans la continuité des langages de programmation logique s'appuyant sur les clauses de Horn. En terme de syntaxe, Kalamar a néanmoins pris le parti de remplacer l'utilisation des symboles (`:-`, `;`, `.`) par des mot-clefs (`if`, `then`, `and`, `or`, etc.) considérés comme plus lisibles à l'usage aussi bien par les ingénieurs de la connaissance que par les clients qu'ils accompagnent. L'ajout de facilités syntaxiques autour des propriétés est également un point essentiel pour une expérience de modélisation améliorée pour les ingénieurs de la connaissance. Elles permettent notamment de distinguer les règles issues de la description d'une propriété (comme la réflexivité) de celles propres aux règles métier du domaine, apportant de la concision dans la description. L'organisation des descriptions des classes et des propriétés par « cadre » ou « *frame* », similaire à la syntaxe Manchester [10], contribue également à la lisibilité des modèles Kalamar en regroupant les informations propres à une même classe au sein d'un seul bloc syntaxique.

4. La définition formelle est omise dans le présent document pour des raisons de lisibilité, et peut être inférée des paramètres de la fonction.

La proposition portée par le W3C avec OWL [15] et SWRL [11] repose sur les contraintes définies au sein des logiques de description [3]. De manière générale, Kalamar ne repose pas sur ces éléments et se restreint au vocabulaire RDF(S) (`rdfs:Class`, `rdf:Property`, ...). En particulier, Kalamar n'offre pas les expressions de classes (union, intersection, complément) et les axiomes de classes (disjonction) que propose OWL. Une correspondance est néanmoins possible (même si limitée) à travers OWL RL, le fragment de OWL représentable par un ensemble de règles. De ce point de vue, Kalamar est à rapprocher de SWRL qui agrémente OWL de structures issues de RuleML. Kalamar reprend à OWL les terminologies des axiomes de propriétés (réflexivité, symétrie, transitivité, inverse) mais conserve une indépendance. Cela permet par exemple d'intégrer d'autres choix de langages de spécification de schéma que OWL, comme par exemple SHACL[5] dont la sémantique exprimée sous forme de règles peut être formalisées par des tests Kalamar (section 3.6).

Notation3 [5] est un langage simplifié pour exprimer les informations portées par RDF, en réduisant la syntaxe à quelques symboles et se fondant sur le triplet comme primitive. Cela permet une écriture plus lisible que la syntaxe XML/RDF souvent présentée comme trop chargée. Notation3 permet aussi d'exprimer des règles existentielles [2], également sous forme de triplets. Cependant, il s'appuie principalement sur quelques symboles (`=>`, `.`, `;`) qui rencontrent la même objection du manque de lisibilité que Prolog et Datalog.

Kalamar est aussi un langage en évolution. Une future version en préparation intégrera de nouvelles facilités d'écriture pour permettre la réification de propriétés, la définition de propriétés par du calcul aboutissant à une règle sous-jacente, ou encore du calcul en respectant les unités de mesure. Ces extensions sont bien entendu supportées par l'expérience.

Ainsi, Kalamar est une syntaxe de haut niveau pour permettre d'exprimer des règles et construire des modèles formels de domaines métier. À la manière de la syntaxe Manchester pour OWL [10], Kalamar propose une syntaxe plus concise et lisible pour l'humain, ingénieur de la connaissance ou client, au-dessus des langages de règles tels que Datalog, Prolog ou SWRL.

Références

[1] Pauline Armary and Brice Sommacal. Cortex : An experimentation for a e-health data hub. Présentation dans le *Industry Track* de Semantics'24.

[2] Dörthe Arndt and Stephan Mennicke. Notation3 as an existential rule language. In *International Joint Conference on Rules and Reasoning*, pages 70–85. Springer, 2023.

[3] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[4] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, Swan Rocher, and Clément Sipieter. Graal : A toolkit for query answering with existential rules. In *Rule Technologies : Foundations, Tools, and Applications*, pages 328–344, Cham, 2015. Springer International Publishing.

[5] Tim Berners-Lee and Dan Connolly. Notation3 (n3) : A readable rdf syntax. *https ://www.w3.org/TeamSubmission/n3/*, 2011.

[6] Harold Boley, Said Tabet, and Gerd Wagner. Design rationale of ruleml : a markup language for semantic web rules. In *Proceedings of the First International Conference on Semantic Web Working*, SWWS'01, page 381–401, Aachen, DEU, 2001. CEUR-WS.org.

[7] Michel Vanden Bossche, Maxime Van Assche, and Carlos Noguera. Extended swrl for commercial-scale ontology-centric applications.

[8] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena : implementing the semantic web recommendations. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, WWW Alt. '04, page 74–83, New York, NY, USA, 2004. Association for Computing Machinery.

[9] Alain Colmerauer, Henry Kanoui, and Michel Van Caneghem. Last steps towards an ultimate PROLOG. In *Proceedings of IJCAI'81, Vancouver, BC, Canada, August 24-28, 1981*, pages 947–948. William Kaufmann, 1981.

[10] Matthew Horridge and Peter Patel-Schneider. OWL 2 web ontology language. manchester syntax (second edition), December 2012. https://www.w3.org/TR/owl2-manchester-syntax/.

[11] H Lan. Swrl : A semantic web rule language combining owl and ruleml. *http ://www. w3. org/Submission/SWRL/*, 2004.

[12] Alejandro Mallea, Marcelo Arenas, Aidan Hogan, and Axel Polleres. On blank nodes. In *International semantic web conference*, pages 421–437. Springer, 2011.

[13] Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. Rdfox : A highly-scalable rdf store. In *The Semantic Web - ISWC 2015*, pages 3–20, Cham, 2015. Springer International Publishing.

[14] André Pacak and Sebastian Erdweg. Functional programming with datalog. *European Conference on Object-Oriented Programming*, 2022.

[15] Peter Patel-Schneider, Elisa Kendall, Deborah McGuinness, and Jie Bao. OWL 2 web ontology language quick reference guide (second edition). Technical report, W3C, December 2012.

[16] Daniel Strmečki, Ivan Magdalenić, and Danijel Radosević. A systematic literature review on the application of ontologies in automatic programming. *International journal of software engineering and knowledge engineering*, 28(05) :559–591, 2018.

5. https://www.w3.org/TR/shacl/

SOLAR-FU : Raisonner avec des règles logiques du second ordre dans une unification de bases de connaissance

David Camarazo[1,2], Mohammed Lalou[2], Ana Roxin[2]

[1] IRT Railenium
[2] Université Bourgogne Europe, LIB UR 7534, 21000 Dijon, France

9 mai 2025

Résumé

Dans des contextes industriels, la modélisation de systèmes complexes implique la production de nombreux modèles dans différents formats. La cohérence de ces modèles est essentielle pour garantir la sécurité et la fiabilité du système. Cependant, la vérification de cette cohérence est une tâche difficile en raison de l'hétérogénéité des modèles. Dans ce travail, nous proposons une approche de vérification semi-automatisée basée sur une unification de bases de connaissances, permettant d'intégrer les connaissances de différents modèles dans une structure cohérente, et un mécanisme de raisonnement utilisant des règles logiques du second ordre permettant la formulation et l'application de règles de vérification réutilisables et généralisables. Nous illustrons notre approche sur un cas d'usage simplifié et discutons son implémentation, ainsi que les perspectives d'évolution vers des applications industrielles plus larges.

Mots-clés

Base de connaissances, Raisonnement, Logique d'ordre deux, Ingénierie des connaissances, Vérification formelle, Ingénierie dirigée par les modèles, Patterns ontologiques, Mécanisme de template

Abstract

In industrial contexts, modelling complex systems involves producing numerous models in different formats. The consistency of these models is essential to guarantee system safety and reliability. However, verifying this consistency is a difficult task due to the heterogeneity of the models. In this work, we propose a semi-automated verification approach based on the unification of knowledge bases, enabling the integration of knowledge from different models into a coherent structure, and a reasoning mechanism using second-order logic rules, enabling the formulation and application of reusable and generalizable verification rules. We illustrate our approach on a simplified use case and discuss its implementation and prospects for evolution towards broader industrial applications.

Keywords

Knowledge base, Reasoning, Second order logic, Knowledge engineering, formal verification, Model driven engineering, Ontology patterns, Template mechanism

1 Introduction

Dans un contexte industriel, la modélisation de systèmes complexes, tels que les systèmes ferroviaires, systèmes aéronautiques, systèmes nucléaires etc., produit de nombreux modèles par des acteurs ayant des connaissances métiers différentes et utilisant des outils différents. Dans ce contexte, pour assurer la viabilité du système produit, il est important, voire critique (si une défaillance du système pose un risque pour la santé des usagers), de vérifier les modèles produits [5]. L'objectif des travaux présentés dans cet article est de proposer une approche qui assiste les acteurs de la vérification en proposant un formalisme qui permette d'organiser, de mutualiser et de traiter les connaissances des différents acteurs de la modélisation. Plus précisément, nous souhaitons spécifier une méthode de vérification semi-automatisée de structure de modèles de systèmes complexes. Pour enregistrer les données produites durant un processus de modélisation de façon explicite et formelle, nous utilisons des bases de connaissances qui sont des structures de données conçues pour répondre à ce besoin [20]. Nous devons alors répondre à deux problématiques pour spécifier notre méthode de vérification, à savoir : comment organiser la connaissance impliquée dans le processus de modélisation ? Et comment interroger la structure de connaissances utilisée dans le processus de vérification ?

Nous répondons à la première problématique en spécifiant une unification de bases de connaissances [23]. Chaque format manipulé durant le processus de modélisation a sa propre base de connaissances indépendante et toutes ces bases sont unifiées par une base de connaissances dite "de vérification" (KBV pour Knowledge Base Verification) contenant les règles de vérification de structure des modèles. Pour faciliter l'écriture des règles de vérification, KBV peut contenir des règles logiques d'ordre deux.

Nous répondons à la seconde problématique en proposant un mécanisme de raisonnement en chaînage arrière qui réécrit la requête d'abord avec les règles logiques d'ordre deux de KBV puis avec les règles logiques d'ordre un présentes dans KBV et dans les autres bases de connaissances.

Nous détaillons notre approche SOLAR-FU (Second Order Logic and Reasoning For Unification) dans les sections

suivantes de cet article. La section 2 introduit les notions importantes que nous utilisons dans le reste de l'article. La section 3 présente les autres travaux qui ont abordé les problèmes de vérification formelle semi-automatisée et positionne notre approche. La section 4 décrit notre approche en détail avec la spécification de l'organisation des connaissances et le mécanisme d'interrogation de la structure des connaissances comportant des règles logiques du second ordre. La section 5 illustre le fonctionnement de notre approche sur un cas d'usage simplifié. La section 6 détaille nos choix concernant l'implémentation. Enfin, la section 7 conclut cet article et présente nos ambitions pour les travaux futurs.

2 Définitions et notations

2.1 Base de connaissances

Pour construire une base de connaissances, on doit d'abord définir certaines notions élémentaires.

On utilise un vocabulaire composé de deux ensembles. Un ensemble de prédicats $\mathcal{P}$ et un ensemble de termes $\mathcal{T}$. Un prédicat a une arité qui correspond au nombre de termes sur lesquels porte le prédicat. On appelle "atome" un prédicat associé à un ensemble de termes. Soit p un prédicat d'arité n et $[t_1, t_2, ..., t_n]$ un ensemble de termes, on note un atome $p(t_1, t_2, ..., t_n)$.

Suivant les définitions de [14], une base de connaissances est une paire composée d'une base de faits et d'une base de règles. La base de faits est une conjonction d'atomes. Dans cet article, on note une conjonction d'atomes en séparant les atomes par des virgules comme ceci $p_1(t_1), p_2(t_1, t_2)$. La base de règles est un ensemble de règles d'ordre un. Une règle du premier ordre est une formule logique du premier ordre de la forme $r = \forall \vec{x}, \vec{y}, C(\vec{x}, \vec{y}) \rightarrow T(\vec{x})$ où $\vec{x}$ et $\vec{y}$ sont des ensembles de termes, C est une conjonction d'atomes et T est un atome. Pour simplifier l'écriture, dans la suite de ce document, on notera les règles en ignorant les quantificateurs et en utilisant le formalisme DataLog [11]. Une règle sera donc écrite sous la forme $r = T(\vec{x}) : -C(\vec{x}, \vec{y})$. . De plus, on appelle le corps d'une règle $r = \forall \vec{x}, \vec{y}, C(\vec{x}, \vec{y}) \rightarrow T(\vec{x})$ la prémisse de l'implication de r i.e $C(\vec{x}, \vec{y})$. De façon analogue,on définit la tête d'une règle $r = \forall \vec{x}, \vec{y}, C(\vec{x}, \vec{y}) \rightarrow T(\vec{x})$ la conséquence de l'implication de r, i.e $T(\vec{x})$. On notera dans la suite du document $corps(r)$ le corps d'une règle r et $tete(r)$ la tête d'une règle r.

On s'intéresse maintenant aux définitions relatives aux requêtes et à leur évaluation sur la base de connaissances. On définit une requête q comme une paire composée d'une conjonction d'atomes qf et d'un ensemble de variables réponses $\vec{v}$ qui apparaissent dans la conjonction d'atomes qf. Dans la suite de ce document, on note les requêtes en utilisant le formalisme Datalog : $?(\vec{v}) : -qf$.

Pour évaluer les requêtes, on définit d'abord un homomorphisme d'une conjonction d'atomes F_1 dans une conjonction d'atomes F_2, comme une substitution σ des variables de F_1 vers les termes de F_2 telle que $\sigma(F_1) \subseteq F_2$. Les homomorphismes permettent d'évaluer les requêtes. Soit q une requête $\vec{v} : -qf$ et une base de faits bf, une substitution σ est une réponse de q dans bf s'il existe un homomorphisme h de qf dans bf tel que $\sigma \subseteq h$. L'évaluation d'une requête q dans bf calcule toutes les substitutions σ qui sont réponses de q dans bf.

Dans cet article, nous utilisons un mécanisme de raisonnement dit "chaînage arrière" pour inférer des informations supplémentaires sur les données. Pour détailler ce processus, nous définissons d'abord un unificateur. Un unificateur u entre une requête q et une règle r est une substitution des variables de q et de r telle que $u(tete(r)) \subseteq u(q)$. On peut alors définir ce qu'est une réécriture de requête. On dit que la requête q' est une réécriture de la requête q par la règle r si et seulement s'il existe un unificateur u tel que $q' = u(q) \setminus u(tete(r)) \cup u(corps(r))$. On définit alors l'opérateur $reecrit(q, regles)$ qui prend en entrée une requête q et un ensemble de règles $regles$ et qui calcule les réécritures de q par les règles $regles$. Cet opérateur reprend l'algorithme de réécriture détaillé dans [14]. Cet opérateur nous permet de réaliser un raisonnement en chaînage arrière, c'est-à-dire de partir d'une requête formulée par l'utilisateur sur une base de faits bf, puis de la réécrire avec l'opérateur $reecrit(q, regles)$ pour interroger bf avec les réécritures de q.

2.2 GBox et générateurs

Dans ce document, on appelle une règle logique du second ordre une règle logique du premier ordre dont certains prédicats sont des prédicats variables, qui peuvent être substitués par des prédicats non variables, qu'on appelle des prédicats concrets. Dans la suite du document, on note SO-Rule (Second Order Rule) pour dire "règle logique du second ordre" ou "règle logique d'ordre deux".

On reprend alors la définition de [12], que l'on adapte au formalisme des bases de connaissances, pour définir un générateur comme une paire de deux SO-Rules, un pattern et un template, tel que tous les prédicats variables du template apparaissent dans le pattern. Dans la suite du document, on note $pattern(g)$ le pattern d'un générateur g et de façon analogue, $template(g)$ le template d'un générateur g.

Avant de définir l'utilisation d'un générateur, on définit un isomorphisme d'une conjonction d'atomes F_1 dans F_2 comme une substitution σ telle que σ associe exactement un terme de F_1 à un terme de F_2 et $\sigma(F_1) = F_2$. On note qu'il existe un isomorphisme entre F_1 et $F_2 : F_1 \equiv F_2$.

On définit maintenant l'expansion d'un générateur sur une base de règles $\mathcal{R}$. L'expansion d'un générateur est l'ensemble de règles correspondant au template auquel on a substitué les prédicats variables par des prédicats concrets. Toutefois, on autorise une substitution σ des prédicats variables vers des prédicats concrets si et seulement si pour toutes les règles $r \in pattern(g), \sigma(r) \equiv r', r' \in \mathcal{R}$. On définit donc l'expansion d'un générateur g comme l'ensemble des règles $expansion = \{\sigma_i(template(g) | \forall r \in pattern(g), \exists r' \in \mathcal{R}, \sigma_i(pattern(g) \equiv r'\}$.

L'intérêt d'un générateur est qu'il permet de représenter des connaissances plus abstraites et de capturer des régularités qui peuvent être difficiles à représenter avec des règles

d'ordre un. Concrètement, un générateur permet de produire à partir de patterns et de templates, un ensemble de règles du premier ordre avec lesquelles il est possible d'utiliser les opérateurs de raisonnement classique pour inférer des connaissances.
On peut par exemple écrire un générateur $chemin(X,Y) : -v\_X(X,Y). => v\_X(X,Z) : -v\_X(X,Y), v\_X(Y,Z).$ qui génère automatiquement une règle de transitivité pour toutes les relations qui sont des sous-relations de la relation $chemin$.

3 Travaux connexes

3.1 Vérification formelle

Pour réaliser une vérification formelle de modèles, les méthodes formelles telles que proposées dans [26] sont très présentes dans la littérature scientifique. Cette approche consiste à transformer les exigences et le comportement du système en une conjonction de formules logiques et de fonctions mathématiques, puis vérifier si les formules sont satisfaites dans n'importe quel état du système. Une ontologie peut être utilisée pour représenter la connaissance des contraintes de sécurité, comme dans [26], mais ce n'est pas toujours le cas.
Une autre approche couramment utilisée est l'annotation pour établir une relation entre un modèle et une ontologie de domaine. Un exemple de cette approche est présenté dans [7]. L'objectif principal est d'aligner un modèle informel avec une ontologie de domaine afin d'appliquer des règles de vérification sur les entités du modèle. Cette approche permet de vérifier la cohérence entre les connaissances représentées par le modèle, souvent de manière implicite et informelle, et celles formalisées dans l'ontologie de domaine.
Enfin, nous identifions une dernière méthode de vérification proposée pour l'aéronautique [18]. Cette méthode repose sur une ontologie pour vérifier les données produites durant le processus de modélisation. L'approche consiste à enregistrer les données de l'architecture du système afin de pouvoir les interroger pour simuler le comportement du système. Les résultats de la simulation sont alors comparés avec les besoins enregistrés dans une ontologie des besoins pour vérifier les données de l'architecture.

3.2 Positionnement dans l'existant

Tout d'abord, la comparaison avec les méthodes formelles classiques [26] fait apparaître une différence de point de vue. Les méthodes formelles sont conçues pour vérifier le comportement du système, en s'assurant qu'aucun état du système ne viole une règle de vérification. Cette approche est conçue pour vérifier que le système ne peut pas atteindre un état dangereux, ce qui n'est pas notre objectif. Notre objectif est de vérifier que les modèles et documents produits dans le processus de modélisation sont cohérents entre eux. Par exemple, les méthodes formelles peuvent vérifier qu'aucun train n'est présent simultanément sur la même section de rail. Cependant, elles sont moins efficaces pour décrire la topographie d'un train (par exemple, combien de wagons il possède, où se trouve la cabine du pilote, quels sont les capteurs qu'il possède et ce qu'ils mesurent ?) Par conséquent, notre travail peut être considéré comme préliminaire et nécessaire à la mise en œuvre de méthodes formelles. En effet, les méthodes formelles nécessitent un modèle complet et approfondi du système qui peut être transformé en formules et fonctions logiques correctes. Tandis que notre méthode vise à assister la phase de modélisation, les méthodes formelles sont conçues pour vérifier et simuler le comportement du système à la fin du processus. En tant que telle, notre approche est complémentaire des approches basées sur les méthodes formelles.
Ensuite, nous comparons notre approche avec l'approche d'annotation décrite dans [7]. Cette approche vise à expliciter la sémantique d'un document en utilisant des balises pour lier le contenu du document à une ontologie spécifiant la définition formelle, lisible par une machine et explicite du contenu du document. L'objectif est de vérifier le contenu du document en rendant sa sémantique explicite et en utilisant un raisonneur pour la vérification sémantique formelle du document (dans notre cas, les documents sont des modèles). Cela ne correspond pas tout à fait à ce que nous proposons. Notre proposition ne vise pas à vérifier l'adéquation du vocabulaire d'un modèle produit durant le processus de modélisation, en vérifiant la sémantique du modèle. On veut vérifier la structure des modèles. Plus précisément, nous souhaitons vérifier l'organisation des données et surtout les liens entre les données produites lors du processus de modélisation.
Enfin, comparons notre proposition avec la proposition de [18]. Cette approche est très similaire à ce que nous essayons de réaliser. Plusieurs ontologies sont unifiées dans le but de vérifier la cohérence entre les données présentes dans les différentes ontologies. Il y a toutefois plusieurs différences notables. Notre approche n'unifie pas par une ontologie de haut niveau, mais directement par l'ontologie qui spécifie les règles de vérification. Notre approche ne spécifie aucun langage de modélisation et la vérification est complètement extérieure aux modèles. L'approche proposée par [18] nécessite d'utiliser l'outil MetaGraph 2.0 et son langage de modélisation pour produire les modèles d'architectures et les règles de vérification en langage Karma (langage supporté par MetaGraph 2.0 pour exprimer des contraintes sur les modèles). De plus, il est nécessaire d'utiliser un logiciel externe, un validateur, pour vérifier les données. Notre approche tire profit des opérateurs de raisonnement des ontologies pour vérifier les données. Par l'ajout de la GBox, par l'externalisation de la vérification aux modèles et la formalisation de la vérification, notre approche peut être vue comme une alternative plus générale à l'approche [18].

4 Spécification de notre approche

4.1 Organisation de la connaissance : le choix de l'unification

Concernant la façon de structurer la connaissance, nous nous appuyons sur les conclusions de [16] et cherchons à travailler sur l'aspect modulaire de la structure des connais-

sances et sur la réutilisation des ressources produites pour la structure des connaissances. En effet, bien que ces aspects aient fait l'objet de travaux antérieurs, comme [19] ou [3], ou de travaux de standardisation, comme [13], nous pensons qu'il est toujours possible d'améliorer les pratiques existantes. Les critères principaux que nous cherchons à améliorer sont donc la modularité (i.e. la facilité à modifier la structure de connaissance) et la réutilisabilité. Dans ce but, nous proposons une structure, dans laquelle on peut intégrer différentes bases de connaissances chacune spécifiant les connaissances relatives à un format utilisé dans le processus de modélisation. On pourra donc utiliser un raisonneur pour interroger les connaissances de chaque document produit. En compartimentant les connaissances, on peut spécifier des mécanismes de formalisation automatique réutilisables pour chaque document du même format e.g. tous les documents de safety peuvent être formalisés pour peupler une ontologie safety en suivant un même processus. Les différentes ontologies sont unifiées [23] par une base de connaissances de vérification (KBV), qui devra être modifiée en fonction des connaissances qu'elle unifie. Pour adresser les problèmes de réutilisabilité des fragments de connaissance développés pour un projet précis, comme ceux soulevés dans [2], nous proposons une KBV avec une GBox. La GBox permet de rajouter un niveau d'abstraction dans l'ontologie de vérification ce qui facilite sa réutilisation. Nous illustrons la structure de connaissances que nous proposons ci-dessous en figure 1.

Nous formulons l'hypothèse que cette structure facilite la réutilisation des ressources et est plus modulaire. Comme vu précédemment, l'unification permet d'ajouter facilement de nouvelles connaissances dans l'ontologie de vérification. KBV utilise une GBox pour faciliter la spécification de règles de vérification. On suit la vision Dogma [4], la base de connaissances de vérification ne fait que spécifier les contraintes sur le vocabulaire des connaissances unifiées. Ce découplage entre le vocabulaire et les contraintes qui sont posées dessus offre un équilibre entre utilisabilité et réutilisabilité. Dans cette configuration, on peut en effet plus facilement réutiliser les terminologies produites. De plus, en utilisant une GBox pour générer les règles de vérification, on suit les conseils des méthodes Neon [24] et Lot [22] qui insistent sur l'importance d'utiliser des ressources plus abstraites pour les rendre plus facilement réutilisables.

Afin d'assurer que le raisonnement termine et tire profit du mécanisme de chaînage arrière, nous posons certaines conditions sur les TBox et GBox considérés. Pour les TBox, nous considérons des règles ne contenant que des conjonctions d'atomes dans le corps et un atome dans la tête. Le corps des règles peut également inclure des atomes négatifs. Toutes les variables sont quantifiées universellement. Nous exposons maintenant les restrictions sur la GBox. D'abord, les templates des GBox ne produisent que des prédicats interrogés par les requêtes. Ensuite, les têtes des templates ne peuvent pas produire des prédicats présents dans les corps des templates.

Bien que cette structure des connaissances soit conçue pour faciliter la réutilisation de tout ou partie de la structure, elle reste, comme toute structure de connaissances, difficile à construire. En effet, la construction d'une structure de connaissance nécessite la concertation de plusieurs acteurs ayant chacun des activités et des connaissances différentes [15]. A minima, il s'agit d'articuler la coopération entre des experts du domaine et des ingénieurs en ingénierie des connaissances [6]. Toutefois, il s'agit là d'une problématique à part entière, différente de la nôtre et qui a déjà été traitée dans de nombreux travaux, comme le montre [16].

4.2 Mécanisme d'interrogation en chaînage arrière avec des règles du second ordre

Pour vérifier automatiquement la structure des modèles, il suffit d'interroger la structure de connaissance précédemment définie. Le raisonneur calcule automatiquement les erreurs présentes dans les données en utilisant les règles de vérification présentes dans les TBox et GBox. Concernant le mécanisme d'interrogation, nous formulons l'hypothèse que seule l'ontologie de vérification est interrogée à l'aide d'une requête. Pour proposer un meilleur mécanisme d'interrogation, nous nous appuyons sur les propositions de travaux futurs de [12] qui pose notamment la question : "Est-il possible de raisonner avec une GBox sans étendre les templates qu'elle contient ?". Nous amorçons avec ces travaux la réflexion sur cette question et présentons un cas où il est possible de raisonner en chaînage arrière avec une GBox. En effet, pour éviter d'étendre les règles de la GBox, on doit utiliser un mécanisme de raisonnement en chaînage arrière qui ne calculera qu'une expansion partielle des règles de la GBox en considérant les prédicats de la requête, i.e qui ne calculera que les expansions dont on a besoin pour répondre à la requête. Dans une optique de vérification, le chaînage arrière se justifie car il correspond exactement à ce qu'on souhaite faire. La requête exprime la vérification souhaitée, et le chaînage arrière produit la connaissance pour opérer la vérification demandée. Enfin, nous faisons l'hypothèse que l'on raisonne en monde fermé. L'hypothèse du monde fermé permet de discriminer les individus qui n'ont pas les propriétés souhaitées, ce qui est pertinent pour une tâche de vérification. Une illustration de ce mécanisme de requête est présentée sur la figure 2.

Maintenant que nous avons établi la structure de connaissances et le mécanisme de raisonnement, nous pouvons spécifier un algorithme pour procéder à la vérification des modèles.

4.3 Algorithme

Pour spécifier notre algorithme, on considère en données d'entrées la requête de vérification q, une base de règles br obtenue par l'union des bases de règles des bases de connaissances de l'unification et bf une base de faits obtenue par l'union des bases de faits des bases de connaissances unifiées. Avec ces éléments d'entrées, nous proposons un algorithme en quatre étapes :

1. Pour chaque générateur de la GBox, on calcule l'expension du template si et seulement si la tête du template produit au moins un des prédicats interrogés. On obtient alors un ensemble de règles du premier

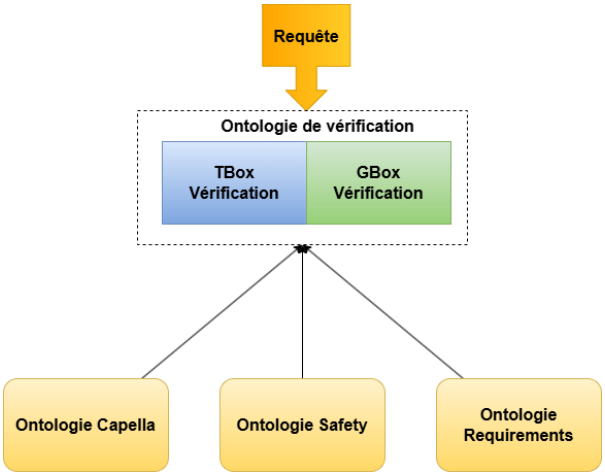

FIGURE 1 – Structure de connaissance

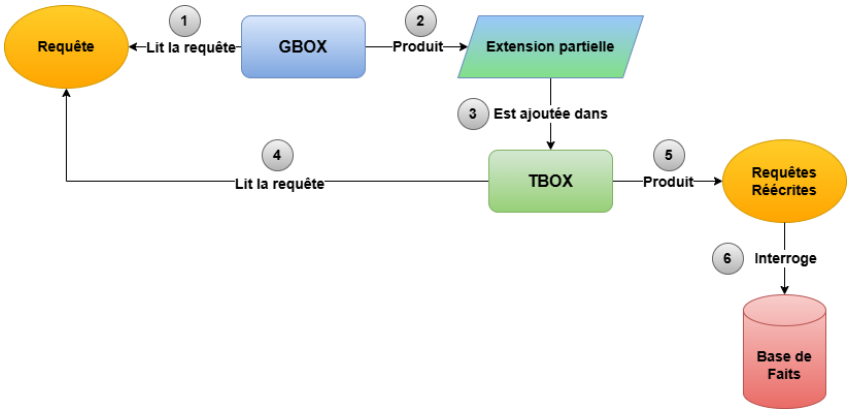

FIGURE 2 – Mécanisme de raisonnement

ordre $regles\_generes$.

2. On réécrit la requête q avec les règles $regles\_generes$ pour obtenir un ensemble de requêtes $reecritures\_partielles$.
3. On réécrit alors les requêtes $reecritures\_partielles$ avec les règles de br pour obtenir un ensemble de requêtes $reecritures$.
4. Enfin, on évalue les requêtes $reecritures$ sur la base de faits bf pour calculer un ensemble de substitutions, $reponses$, des variables réponses de q vers les constantes de bf pour répondre à la requête q.

Pour calculer la complexité de cet algorithme, on note G la taille de la GBox en nombre de générateurs, R la taille de br en nombre de règles et F la taille de la base de faits en nombre d'atomes. On a une complexité pour le calcul de l'expansion des générateurs en $O(G \times R)$. En effet, on parcourt une fois tous les générateurs, et pour chaque générateur, on parcourt une fois toute la base de règles pour calculer les substitutions. Pour calculer la complexité de l'algorithme dans sa globalité, on ajoute à la complexité du calcul de l'expansion la complexité des réécritures, $O(R + R^2)$. En effet, on parcourt l'expansion des générateurs, puis pour chaque requête réécrite avec l'expansion, on parcourt la base de règles pour la réécrire avec br. Puisqu'avec les générateurs on génère au maximum R règles, qui peuvent générer chacune une réécriture, la complexité de la réécriture est $O(R + R^2)$. Enfin on y ajoute la complexité de l'évaluation des requêtes qui est $O(R \times F^F)$, puisque la taille maximale des requêtes réécrites est de F, que nous avons au maximum R réécritures et que chaque atome des requêtes réécrites est évalué sur la base de faits de taille F. On a donc une complexité totale en $O(R(G + 1 + R + F^F))$. Il s'agit toutefois ici d'une estimation haute, qui devrait rarement être atteinte en pratique (la taille des requêtes réécrites est souvent très inférieure à la taille de la base de faits, et le nombre de réécritures très inférieur à la taille de la base de règles).

Pour fonctionner, cet algorithme utilise trois opérateurs :

1. $calcule\_substitutions(p, br)$ qui pour un pattern p et un ensemble de règles br, renvoie l'ensemble des substitutions $substitutions$ des prédicats variables de p vers les prédicats concrets de br tels que $\forall \sigma_i \in S, \exists r \in br, \sigma_i(p) \equiv r$.
2. $reecrit(q, br)$ qui pour une requête q est un ensemble de règles br calcule l'ensemble des réécri-

Données : bf une base de faits, br une base de règles, q une requête
Résultat : $reponses$ la liste des réponses à la requêtes q
$regles\_generes \leftarrow []$;
pour chaque *générateur* $g \in GBox$ **faire**
  $expansion \leftarrow []$;
  pour chaque *Règle* $sorule \in template(g)$ **faire**
    si *tête de* $sorule$ *a un prédicat présent dans* q **alors**
      $substitutions \leftarrow calcule\_substitutions(pattern(g), br)$;
      pour chaque $\sigma \in substitutions$ **faire**
        $expansion \leftarrow expansion \cup \sigma_i(sorule)$;
      fin
    fin
  fin
  $regles\_generes \leftarrow regles\_generes \cup expansion$;
fin
$reecritures\_partielles \leftarrow reecrit(q, regles\_generes)$;
$reecritures \leftarrow []$;
pour chaque *requête* $q' \in reecritures\_partielles$ **faire**
  $reecritures \leftarrow reecritures \cup reecrit(q', br)$;
fin
$reponses \leftarrow []$;
pour chaque *requête* $q'' \in reecritures$ **faire**
  $reponses \leftarrow reponses \cup eval(q'', bf)$;
fin
retourner $reponses$

Algorithme 1 : Algorithme synthétisant l'approche SOLAR-FU

tures de q avec les règles de br.

3. $eval(q, bf)$ qui pour une requête q est une base de faits bf calcule l'ensemble des substitutions $reponses$ tels que $\forall \sigma_i \in reponses, \sigma_i(q) \subseteq bf$.

L'algorithme et les opérateurs présentés dans cette section permettent de faire fonctionner notre approche sur des jeux de données réels. Un exemple est présenté dans la section suivante.

5 Évaluation conceptuelle de l'approche

Afin d'illustrer le fonctionnement de notre approche, nous détaillons dans cette section son déroulé sur un cas d'usage. Nous utilisons dans cette section les notations DataLog avec les noms de variables en majuscule et les noms des prédicats commencent par une minuscule. Supposons que l'on ait d'un côté des modèles SysML [8] représentant les décompositions fonctionnelles d'un système, et de l'autre des tableurs excel détaillant les exigences auxquelles doit répondre le système. Les modèles SysML sont des diagrammes représentant graphiquement les composants du système et les fonctions qu'ils réalisent. Les tableurs spécifient pour chaque exigence, la fonction d'un modèle SysML qui lui est associée à l'aide d'un identifiant de fonction. Les fonctions et les exigences sont réparties sur deux couches de modélisation, la couche logique qui spécifie des fonctions et exigences haut niveau sans détailler la technologie utilisée, et la couche physique qui précise les technologies utilisées. On souhaite vérifier qu'une exigence logique soit toujours associée à une fonction logique, et de façon analogue, qu'une exigence physique soit systématiquement associée à une fonction physique.

Pour ce faire, nous construisons une base de faits pour les exigences $bf_{exigence} = exigenceLog(e1)$, $exigenceLog(e2)$, $exigenceLog(e3)$, $aFonction(e1, a)$, $aFonction(e3, ab)$. Cette base de faits contient les prédicats $exigenceLog$, qui spécifient les instances d'exigences logiques, et $a\_fonction$ qui spécifie le lien entre une exigence et un code de fonction. Nous construisons ensuite la base de faits pour les fonctions $bf_{fonction} = fonctionLog(f1)$, $fonctionLog(f2)$, $fonctionPhy(f3)$, $aCode(f1, a)$, $aCode(f2, b)$, $aCode(f3, ab)$. Cette base de faits contient les prédicats $fonctionLog$, qui spécifient les instances de fonctions logiques, $fonctionPhy$ qui spécifie les instances de fonctions physiques et a_code qui associe à une fonction son code afin de l'identifier.

Pour faire le lien entre les deux bases de faits, on écrit une règle dans la base de connaissances de vérification $aFonctionSysML(X, Y) :- aCode(Y, Z), aFonction(X, Z)$. . Cette règle spécifie que X a pour fonction Y si Y a pour identifiant Z et X est associé au code de fonction Z. Cela permet d'exprimer formellement le lien entre les faits exprimés par les tableaux d'exigences et les faits exprimés par les diagrammes de fonctions. On écrit également deux règles pour exprimer les relations entre la vision logique et la vision physique du

système : $couverture(X,Y) :- exigenceLog(X), fonctionLog(Y).$ et $couverture(X,Y) :- exigencePhy(X), fonctionPhy(Y).$.
On peut alors écrire deux règles de vérification : $mauvaiseAttrib(X,Y) :- exigenceLog(X), aFonctionSysML(X,Y), not\ fonction_logique(Y).$ et $mauvaiseAttrib(X,Y) :- exigencePhy(X), aFonctionSysML(X,Y), not\ fonctionPhy(Y).$. On peut capturer les régularités de ces règles de vérification à l'aide d'un générateur :$couverture(X,Y) :- v\_X(X), v\_Y(Y). => mauvaiseAttrib(X,Y) :- v\_X(X), aFonctionSysML(X,Y), not\ v\_Y(Y).$
Précisons que dans notre implémentation, les prédicats variables sont indiqués par les caractères $v\_$ devant le nom du prédicat variable. Ici on a donc deux prédicats variables, v_X et v_Y. Notons également que dans cet exemple simplifié, la régularité ne porte que sur deux règles et n'est donc pas très intéressante. Toutefois, dans la réalité, il peut exister au moins cinq couches de modélisation (opérationnelle, système, logique, physique, décomposition produit) qui peuvent se décliner en sous-couches. En pratique, il est donc intéressant d'utiliser des templates et patterns de règle pour capturer les règles similaires qui reviennent sur toutes les couches de la modélisation.
On peut alors interroger notre structure de connaissances sur le prédicat $mauvaiseAttrib$ avec la requête suivante $?(X,Y) : -mauvaiseAttrib(X,Y)$..
L'algorithme 1 calcule alors l'expansion du générateur, l'ensemble de règles $regles\_generes$.

```
mauvaiseAttrib(X,Y) :-
exigenceLog(X),
aFonctionSysML(X,Y),
not fonctionLog(Y).
mauvaiseAttrib(X, Y) :-
exigencePhy(X),
aFonctionSysML(X,Y),
not fonctionPhy(Y).
```

Nous passons maintenant à la deuxième étape de l'algorithme 1. Cette étape, illustrée par la figure 3, concerne la réécriture de la requête par les règles générées à l'étape précédente. On réécrit la requête initiale avec les règles générées par l'expansion des générateurs. Ici, on a deux requêtes :

```
?(X,Y) :-  exigenceLog(X),
aFonctionSysML(X,Y),
not fonctionLog(Y).
?(X,Y) :- exigencePhy(X),
aFonctionSysML(X,Y),
not fonction_Phy(Y).
```

On détaille maintenant la troisième étape, illustrée par la figure 4, de l'algorithme 1. On réécrit maintenant les requêtes calculées à l'étape précédente avec les règles de la base de règles comportant les règles de la base de connaissances de vérification et les règles des bases de connaissances unifiées de modèles. On obtient un nouvel ensemble de requêtes. Pour simplifier, nous ne présentons que les requêtes les plus réécrites.

```
?(X,Y) :-  exigenceLog(X),
aFonction(X,Z),
not fonctionLog(Y),
aCode(Y,Z).
?(X,Y) :- exigencePhy(X),
aFonction(X,Z),
not fonctionPhy(Y),
aCode(Y,Z).
```

Enfin on passe à la quatrième et dernière étape de l'algorithme 1 illustrée par la figure 5. On interroge la base de faits de notre structure de connaissances (qui est la conjonction des bases de faits des ontologies de modèles) avec les requêtes produites à l'étape précédente. On obtient alors les réponses à notre requête initiale sous la forme de substitution de variables réponses par des constantes de la base de faits. Si on considère la base de faits illustrée dans la figure 5, on a comme réponses `[{X:e3, Y:f3}]`
En testant cette approche sur une base de faits contenant 1371 atomes, l'algorithme renvoie les réponses en environ une seconde. Sur un exemple aussi simple, une fois l'expansion des générateurs calculée, la stratégie de raisonnement utilisée, chaînage avant ou arrière, n'impacte pas le temps d'exécution. L'algorithme renvoie toujours les réponses en environ une seconde. Il serait difficile de présenter une stratégie de raisonnement optimale pour n'importe quel ensemble de règles et de faits. En effet, comme montré dans [10] et [1], les résultats de ces stratégies varient de façon significative en fonction des règles et faits considérés. À ce stade, il est difficile d'anticiper tous les cas d'utilisation de notre approche, et d'en conclure une stratégie de raisonnement optimale en toutes circonstances. Toutefois, nous espérons dans nos futurs travaux, expérimenter notre approche sur davantage de cas d'usage différents, ce qui nous permettrait de déduire des heuristiques pour utiliser au mieux le chaînage avant et arrière de façon appropriée.
Cependant, cet exemple simple permet déjà d'illustrer les apports importants de notre approche. Si l'on souhaite faire la même chose avec les autres méthodes présentées dans la section 3, plusieurs problèmes apparaissent. Premièrement, il aurait fallu plusieurs règles de vérification similaires au lieu d'écrire un seul générateur. De plus, si l'on rajoute des règles de taxonomie comme $couverture(X,Y) :- exigenceSysteme(X), fonctionSysteme(Y).$ on ne doit pas oublier d'écrire la règle de vérification $mauvaiseAttrib(X,Y) :- exigenceSysteme(X), aFonctionSysML(X,Y), not\ fonctionSysteme(Y).$ qui va avec. Notre approche, basée sur les générateurs, génère la nouvelle règle de vérification automatiquement, limitant donc les erreurs possibles. Enfin, en modifiant le générateur, on peut le réutiliser dans un autre projet, ou un autre contexte. Cela nécessite moins de travail que de modifier les règles une par une.

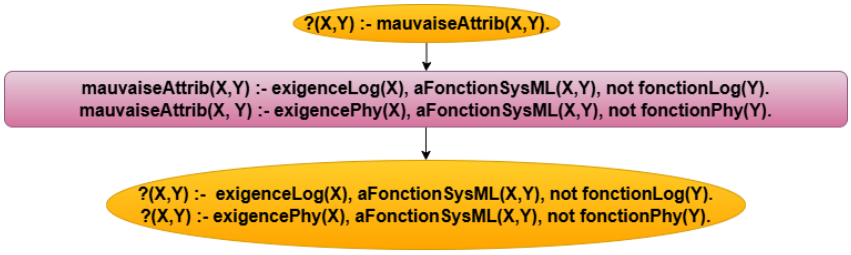

FIGURE 3 – Calcul de la réécriture des requêtes par l'expansion du générateur

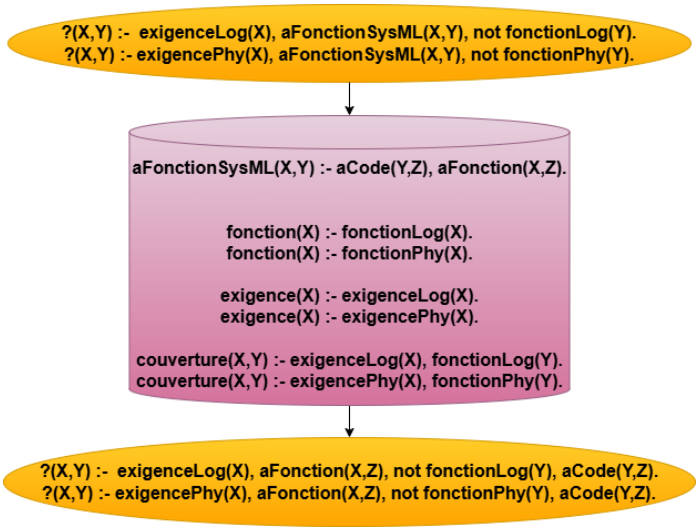

FIGURE 4 – Calcul de la réécriture finale des requêtes

La prochaine section présente nos choix pour implémenter notre approche.

6 Implémentation

Pour réaliser notre implémentation, disponible sur le gitLabhttps://gitlab.com/c4m4r4z0/solar-fu/-/tree/master?ref_type=heads, nous avons d'abord étudié les outils permettant de travailler avec des templates et patterns ontologiques. Une analyse de la littérature révèle que les outils permettant d'exploiter des patterns ontologiques ou des templates ontologiques sont rares. Nous avons malgré tout pu isoler cinq outils qui permettent de manipuler des templates ontologiques :

- GDOL [17] : Generic Distributed Ontology Modeling (GDOL) est un langage proche d'OTTR pour éditer des templates ontologiques. L'article mentionne qu'un plug-in pour Protégé afin d'exploiter ce langage est prévu dans de futurs travaux. Pour l'instant il n'existe à notre connaissance aucun outil accessible permettant de manipuler GDOL.
- Lutra : Lutra est un outil qui utilise le langage OTTR pour éditer des templates. Il permet également d'instancier des templates en interrogeant différentes ontologies ce qui le rend pertinent pour les alignements[19].
- Protégé avec le plug-in OPPL : Ontology Pre-Processing Language (OPPL) [25] est un plug-in de l'éditeur d'ontologie Protégé qui permet de manipuler des patterns ontologiques. Malheureusement, ce plug-in n'est plus mis à jour et nous avons constaté des dysfonctionnements avec les versions actuelles de Protégé.
- Populous : Populous [9] est un outil qui utilise des templates ontologiques et des scripts OPPL pour peupler rapidement des ontologies. L'outil ne semble pas être mis à jour et fonctionne avec une version obsolète du langage OPPL.
- ROBOT : ROBOT est un outil qui permet d'éditer des ontologies. Il propose une fonctionnalité "Template" qui permet de transformer des axiomes exprimés dans un fichier csv en axiome OWL.

Aucune de ces approches ne permet de travailler avec des ontologies contenant des prédicats variables sans étendre la GBox comme cela est relevé dans [12]. De plus, aucune de ces approches ne supporte le chaînage arrière ou l'hypothèse du monde fermé. Nous avons donc préféré étendre le framework Java Integraal https://gitlab.inria.fr/rules/integraal pour implémenter notre approche. Ce framework en libre accès implémente déjà les éléments essentiels de la logique du premier ordre (i.e. atome, prédicat, formule etc.). De plus, il propose de nombreuses fonctionnalités (algorithme de chaînage avant et ar-

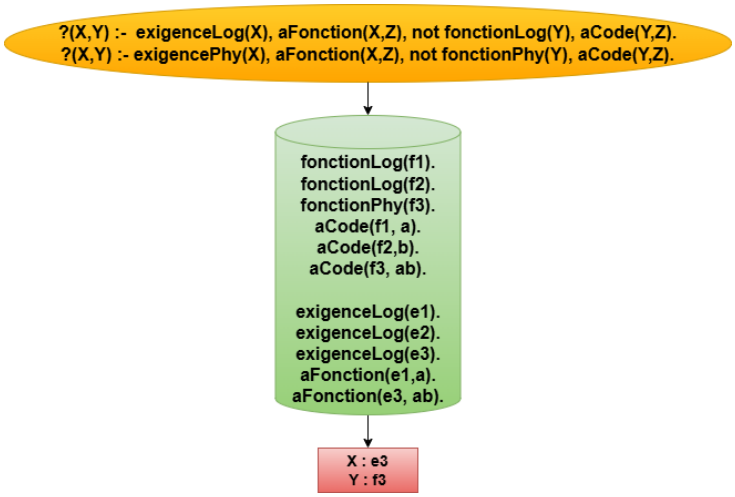

FIGURE 5 – Évaluation des requêtes

rière et évaluation de requête en monde clos) qui sont utiles pour implémenter notre approche.

Nous avons testé cette implémentation sur un jeu de données de la SNCF. Ce jeu de données répertorie 554 fonctions d'un nouveau système de la SNCF dans le cadre du projet Telli [21]. Nous avons utilisé notre approche pour identifier les fonctions qui n'étaient pas liées à une exigence sur la bonne couche de modélisation, ainsi que les fonctions orphelines (qui n'ont pas d'entrée ou de sortie vers une autre fonction). Nous avons identifié avec succès tous les individus attendus, que nous avons d'abord repérés manuellement, dans des temps variant entre 500 et 1000 millisecondes. La mise en place d'un protocole d'évaluation plus rigoureux et sur des données plus importantes fait partie de nos travaux futurs.

7 Conclusion et travaux futurs

Dans le cadre de nos travaux, on s'intéresse à la vérification de la cohérence de modèles avec des formats différents durant la modélisation de systèmes complexes critiques en termes de sécurité. La modélisation de tels systèmes impose un processus formel (pour garantir la sécurité) et rigoureux (pour faciliter la gestion des nombreuses ressources hétérogènes pour modéliser un système complexe).

Nous proposons un processus de vérification centré sur une unification de bases de connaissances. Les différentes bases de connaissances représentant les connaissances relatives aux différents documents utilisés sont unifiées par une base de connaissances de vérification.

Pour faciliter la réutilisation des ressources et l'écriture des règles de vérification, on propose une ontologie de vérification avec une GBox qui contient des générateurs. Ces générateurs sont des paires composées d'un pattern et d'un template. Plus précisément, de deux SO-Rules, c'est-à-dire contenant des prédicats variables.

Pour interroger cette structure, on identifie les générateurs qui permettent d'inférer les atomes pertinents pour la requête. On calcule ensuite les substitutions des prédicats variables qui permettent de générer des règles d'ordre un à partir des templates des générateurs identifiés. On réécrit alors la requête avec les règles précédemment générées. Enfin, on réécrit la requête avec l'ensemble des règles contenues dans les bases de règles des bases de connaissances de l'unification.

Pour nos travaux futurs, nous projetons d'évaluer notre approche sur un nombre de règles et d'instances plus important. Nous prévoyons également d'enrichir notre formalisme en proposant des générateurs qui possèdent chacun plusieurs templates et plusieurs patterns. De plus, nous voulons expérimenter différentes stratégies de raisonnement, en mélangeant le chaînage et le chaînage arrière. Enfin, nous souhaitons adapter notre algorithme pour qu'il puisse fonctionner dans le cas où le raisonnement nécessiterait l'expansion successive de plusieurs générateurs.

Références

[1] Ajlan Al-Ajlan. The comparison between forward and backward chaining. *International Journal of Machine Learning and Computing*, 5(2) :106, 2015.

[2] Valentina Anita Carriero, Marilena Daquino, Aldo Gangemi, Andrea Giovanni Nuzzolese, Silvio Peroni, Valentina Presutti, and Francesca Tomasi. The landscape of ontology reuse approaches. In *Applications and practices in ontology design, extraction, and reasoning*, pages 21–38. IOS Press, 2020.

[3] Laura Daniele, Frank den Hartog, and Jasper Roes. Created in close interaction with the industry : the smart appliances reference (saref) ontology. In *International Workshop Formal Ontologies Meet Industries*, pages 100–112. Springer, 2015.

[4] Aldo De Moor, Pieter De Leenheer, and Robert Meersman. Dogma-mess : A meaning evolution support system for interorganizational ontology engineering. In *Conceptual Structures : Inspiration and Application : 14th International Conference on Concep-*

tual Structures, ICCS 2006, Aalborg, Denmark, July 16-21, 2006. Proceedings 14, pages 189–202. Springer, 2006.

[5] Sana Debbech, Philippe Bon, and Simon Collart-Dutilleul. Improving safety by integrating dysfunctional analysis into the design of railway systems. *WIT Transactions on The Built Environment*, 181 :399–411, 2018.

[6] Christophe Debruyne and Robert Meersman. Gospl : A method and tool for fact-oriented hybrid ontology engineering. In *Advances in Databases and Information Systems : 16th East European Conference, ADBIS 2012, Poznań, Poland, September 18-21, 2012. Proceedings 16*, pages 153–166. Springer, 2012.

[7] Michael Fellmann, Frank Hogrebe, Oliver Thomas, and Markus Nüttgens. An ontology-driven approach to support semantic verification in business process modeling. In *Modellierung betrieblicher Informationssysteme (MobIS 2010). Modellgestütztes Management*, pages 99–110. Gesellschaft für Informatik eV, 2010.

[8] Matthew Hause et al. The sysml modelling language. In *Fifteenth European systems engineering conference*, volume 9, pages 1–12, 2006.

[9] Simon Jupp, Matthew Horridge, Luigi Iannone, Julie Klein, Stuart Owen, Joost Schanstra, Katy Wolstencroft, and Robert Stevens. Populous : a tool for building owl ontologies from templates. *BMC bioinformatics*, 13(1) :1–12, 2012.

[10] Namarta Kapoor and Nischay Bahl. Comparative study of forward and backward chaining in artificial intelligence. *International journal of engineering and computer science*, 5(4) :16239–16242, 2016.

[11] Bas Ketsman, Paraschos Koutris, et al. Modern datalog engines. *Foundations and Trends® in Databases*, 12(1) :1–68, 2022.

[12] Christian Kindermann, Daniel P Lupp, Uli Sattler, and Evgenij Thorstensen. Generating ontologies from templates : A rule-based approach for capturing regularity. In *Description Logics*, 2018.

[13] Wilhelm Klüwer, Johan, Francisco Martin-Recuerda, Daniel Lupp, Arild Waaler, Maja Brandt, Milicic, Stephan Grimm, Aneta Koleva, Mesbah Kahn, Lillian Hella, and Nils Sandsmark. (industrial automation systems and integration — integration of life-cycle data for process plants including oil and gas production facilities — part 14 : Industrial top-level ontology). 2020.

[14] Mélanie König, Michel Leclère, Marie-Laure Mugnier, and Michaël Thomazo. Sound, complete and minimal ucq-rewriting for existential rules. *Semantic Web*, 6(5) :451–475, 2015.

[15] Konstantinos Kotis and George A Vouros. Human-centered ontology engineering : The hcome methodology. *Knowledge and Information Systems*, 10 :109–131, 2006.

[16] Konstantinos I Kotis, George A Vouros, and Dimitris Spiliotopoulos. Ontology engineering methodologies for the evolution of living and reused ontologies : status, trends, findings and recommendations. *The Knowledge Engineering Review*, 35 :e4, 2020.

[17] Bernd Krieg-Brückner and Till Mossakowski. Generic ontologies and generic ontology design patterns. In *WOP@ ISWC*, 2017.

[18] Jinzhi Lu, Junda Ma, Xiaochen Zheng, Guoxin Wang, Han Li, and Dimitris Kiritsis. Design ontology supporting model-based systems engineering formalisms. *IEEE Systems Journal*, 16(4) :5465–5476, 2021.

[19] Daniel P Lupp, Melinda Hodkiewicz, and Martin G Skjæveland. Template libraries for industrial asset maintenance : A methodology for scalable and maintainable ontologies. In *CEUR Workshop Proceedings*, volume 2757, pages 49–64. Technical University of Aachen, 2020.

[20] Nathalie Mitton, Ludovic Brossard, Tassadit Bouadi, Frédérick Garcia, Romain Gautron, Nadine Hilgert, Dino Ienco, Christine Largouët, Evelyne Lutton, Véronique Masson, et al. Fondements et état de l'art. *Archive du CIRAD*, 2022.

[21] G Petitet, M Sango, and P Guicheney. Application de l'ingénierie système pour le renouveau des petites lignes. volume 324, page 44 – 64, 2023.

[22] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. Lot : An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111 :104755, 2022.

[23] International standards development organization. Advanced automation technologies and their applications — requirements for establishing manufacturing enterprise process interoperability. 2015.

[24] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2011.

[25] Ondřej Šváb-Zamazal, Vojtěch Svátek, and Luigi Iannone. Pattern-based ontology transformation service exploiting oppl and owl-api. In *Knowledge Engineering and Management by the Masses : 17th International Conference, EKAW 2010, Lisbon, Portugal, October 11-15, 2010. Proceedings 17*, pages 105–119. Springer, 2010.

[26] Chibuzo Ukegbu, Ramesh Neupane, and Hoda Mehrpouyan. Ontology-based framework for boundary verification of safety and security properties in industrial control systems. In *Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference*, pages 47–52, 2023.

Session 5 : Cadres ontologiques

Ontologies épistémiques *vs* référentielles

Gilles 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 l'étude et la définition d'une classe d'ontologies que nous avons qualifiées d'« épistémiques ». Le rôle assigné à une ontologie épistémique est de rendre compte de connaissances du monde plutôt que du monde directement, a contrario donc du rôle dévolu aux ontologies « référentielles » couramment définies en Ontologie appliquée. Les ontologies épistémiques se distinguent des ontologies référentielles par la nature de leurs catégories et les principes métaphysiques mobilisés pour leur structuration. Dans l'article, nous précisons les différentes notions de conceptualisation qui les sous-tendent puis nous exposons quelques engagements de base que nous adoptons pour le développement des ontologies épistémiques.

Mots-clés

Ontologie épistémique, ontologie référentielle, conceptualisation, représentation mentale, objet de représentation, représentation des connaissances

Abstract

In this article, we continue our study and definition of a class of ontologies we have termed "epistemic". The role assigned to an epistemic ontology is to account for knowledge of the world rather than of the world directly, in contrast to the role assigned to "referential" ontologies commonly defined in Applied Ontology. Epistemic ontologies are distinguished from referential ontologies by the nature of their categories and the metaphysical principles mobilized to structure them. In this article, we define the various conceptualization notions that underlie them and then outline some of the basic commitments we adopt for the development of epistemic ontologies.

Keywords

Epistemic ontology, referential ontology, conceptualization, mental representation, object of representation, knowledge representation

1 Introduction

Dans cet article, nous poursuivons l'étude et la définition d'une classe d'ontologies que nous avons qualifiées d'« épistémiques » [12]. Le rôle assigné à une ontologie épistémique est de rendre compte de connaissances du monde plutôt que du monde directement, a contrario donc du rôle dévolu aux ontologies « référentielles » couramment définies en Ontologie appliquée [3]. Les ontologies épistémiques se distinguent des ontologies référentielles par la nature de leurs catégories et les principes métaphysiques mobilisés pour leur structuration. L'objet de l'article est de clarifier ce qui les différentie.

Dans les domaines philosophiques de l'Ontologie et de la Métaphysique (que nous désignons dorénavant du terme « métaphysique »), les ontologies produites sont classiquement des *systèmes structurés d'entités supposées existantes, organisées en catégories et relations*. En Ontologie appliquée, pour satisfaire à la contrainte que les ontologies soient des composants de systèmes d'information et qu'elles permettent de réaliser des inférences, l'emphase est mise sur la *spécification* compréhensible par ordinateurs des ontologies. Pour traduire cet objectif, les ontologies « computationnelles » sont définies comme des « spécifications d'une conceptualisation », selon la définition désormais classique et très générale de Thomas R. Gruber [7]. Les modes de *spécification* sont variés, à l'instar de ceux des algorithmes, avec des enjeux importants comme ceux de *précision* et de *correction* notamment pour les spécifications logiques vis-à-vis d'une conceptualisation visée (leur *engagement ontologique*). C'est toutefois sur la notion de *conceptualisation* que nous faisons porter notre attention dans cet article. En Ontologie appliquée, les catégories des ontologies – propriétés et relations – correspondent à des universaux instanciés par des individus particuliers. En promouvant en revanche la notion d'ontologie épistémique, nous rompons avec cette orthodoxie en assimilant ces catégories à des objets mentaux généraux pensés par un sujet.

Dans la suite de l'article, nous commençons par distinguer les notions de *conceptualisation* sous-tendant les ontologies référentielles et épistémiques (§§ 2-3), puis nous exposons quelques engagements de base que nous adoptons pour le développement d'ontologies épistémiques de référence (§§ 4-5).

2 Notions abstraite *vs* mentale de *conceptualisation*

Dans un souci de clarifier la signification de termes tels « ontologie », « conceptualisation » et « engagement ontologique », Nicola Guarino et Pierdaniele Garietta [9, p. 31] ont proposé d'assimiler une conceptualisation à « une structure intensionnelle qui code les règles implicites contraignant la structure d'une pièce de la réalité ». Une des motivations de

G&G est de rompre avec une notion extensionnelle de la conceptualisation présente chez Gruber [*ibid.*], assimilant les propriétés et relations à des états de choses (des situations particulières du monde). Pour s'abstraire d'états de choses particuliers, G&G optent pour des propriétés et relations intensionnelles, en faisant appel à une sémantique à la Richard Montague mobilisant une théorie des *mondes possibles*. Les détails de la structure intensionnelle proposée ne sont pas importants, ce qu'il l'est en revanche est de constater que G&G s'engagent vis-à-vis d'une théorie philosophique des concepts – propriétés et relations – les assimilant à des entités abstraites idéales, distinctes de représentations mentales entretenues par des sujets particuliers. Un tel engagement est également celui de Barry Smith [32]. Critiquant la dimension mentale de la notion d'ontologie telle que défendue par Gruber, par exemple dans [8] « Une ontologie est une description (comme une spécification formelle d'un programme) des concepts et relations pouvant exister pour un agent ou une communauté d'agents »[1], Smith condamne le recours à des entités qui ne soient que des « substituts » d'objets existant réellement pour plébisciter des conceptualisations reflétant directement une réalité objective [*ibid.*, p. 163] « C'est précisément parce que de bonnes conceptualisations sont transparentes vis-à-vis de la réalité qu'elles ont une chance raisonnable d'être intégrées ensemble de façon robuste dans un système ontologique unitaire ». Ces critiques unanimes des conceptions de Gruber conduisent à une position très largement majoritaire en Ontologie appliquée consistant à assimiler les catégories d'une conceptualisation à des universaux aristotéliciens [3]. Une telle position rejoint une position également dominante en métaphysique, installée par une lignée de philosophes mathématiciens au 20ème siècle – Frege, Husserl, Carnap, Tarski, Kripke, Montague – qui ont instauré une sémantique idéale. De fait, les termes « conceptuel » et « cognitif » utilisés en ontologie appliquée, par exemple en dénommant « domaine cognitif d'entités » le domaine de discours d'une conceptualisation, ne sont que des emprunts à la psychologie. Les entités de domaines sont considérées comme des objets « cognitivement pertinents » mais restent bien des objets en soi transcendant l'esprit humain – d'où la dénomination d'« ontologie référentielle ».

Un inconvénient immédiat de cette notion de conceptualisation est de ne pas pouvoir rendre compte de la connaissance suivante : en 1859, pour expliquer les perturbations de l'orbite de Mercure, l'astronome français Urbain Le Verrier, célèbre pour ses travaux, postula l'existence d'une planète qu'il nomma Vulcain, censée orbiter entre le Soleil et Mercure. La raison de cette impossibilité en est qu'il est avéré que la planète Vulcain n'existe pas dans le monde physique réel. Il ne peut donc lui correspondre un objet puisque, par définition, un objet représente une entité existante réelle. Les domaines de discours des ontologies référentielles comportent des entités ayant existé dans le passé, voire existant possiblement, et des entités considérées comme existant de façon abstraite, mais toutes existent, ce qui n'est pas le cas d'une entité non-existante. Un autre inconvénient est de devoir souscrire à l'existence des universaux. La notion de conceptualisation ne peut donc contenter les nominalistes des universaux, voire ceux qui, comme l'auteur de l'article, doutent de leur existence. La question est dès lors de savoir si une alternative de cette notion de conceptualisation est envisageable. C'est justement une telle alternative que nous proposons avec la notion d'*ontologie épistémique*.

Pour arrêter cette notion de conceptualisation, nous étendons au mental le cadre ontologique ordinairement établi, pour rendre compte de la capacité que nous confèrent nos représentations à nous référer au monde (l'*intentionnalité*) et à nous forger des jugements sur le monde, bref à nous doter de croyances sur le monde. À cette fin, nous retenons une théorie de l'intentionnalité défendue au tournant du XX^{e} siècle dans l'école brentanienne par Kazimierz Twardowski avec sa théorie des représentations conceptuelles [35,36]. Une telle conception de la représentation mentale s'avère s'accorder avec des travaux contemporains menés en Philosophie de l'esprit et du langage sur les *dossiers mentaux* [21].

La théorie twardowskienne repose sur deux thèses (psychologiques et ontologiques) principales. D'une part, toute représentation fait « advenir » un objet immanent, y compris les représentations n'admettant pas de référence. Twardowski s'oppose ainsi fameusement à la conception des représentations *anobjectuelles* (car non référentes) de Bolzano : « La confusion commise par les défenseurs des représentations sans objet consiste en ceci qu'ils ont tenu la non-existence d'un objet de représentation pour un non-devenir-représenté. Or, toutefois, par chaque représentation, un objet devient représenté, qu'il existe ou non, de même que chaque nom nomme un objet, sans avoir égard au fait que celui-ci existe ou non » [35, § 5]. Twardowski distingue l'objet de sens de l'objet de référence. Par la suite, nous nommons « objet pensé » cet objet immanent de la représentation. D'autre part, dans tout acte de représentation est présent un concept véhiculant comme contenu un objet (l'objet pensé) doté de propriétés [36]. Un concept permet à un sujet de se projeter mentalement un objet relevant d'une catégorie arbitraire, par exemple un objet physique. La structure de la représentation se résume ainsi : acte / concept [objet doté de propriétés] / objet de référence. Toujours selon Twardowski, deux espèces d'objets pensés immanents existent, à savoir des objets *singuliers* (ex : 'Simba', 'la Tour Eiffel, 'la pneumonie de Paul') et des objets *généraux* (ex : 'un lion', 'un bâtiment', 'une pneumonie'). Les premiers représentent des individus particuliers, tandis que les derniers représentent indirectement des individus en possédant des propriétés communes à une pluralité d'objets singuliers. Les objets généraux sont organisés en une hiérarchie. On peut dès lors considérer que chaque sujet pensant dispose d'une ontologie d'objets généraux pensés [14].

En résumé, concernant notre notion de conceptualisation, nous retenons la notion philosophique rappelée en Introduction d'un

[1] Une synthèse des définitions proposées par Gruber est consultable en ligne sur la page « What is an Ontology? » : http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

« système structuré d'entités supposées existantes, organisées en catégories et relations » en assimilant les catégories (et relations) à des objets généraux de pensée représentant le monde.

3 Concept *vs* conception

En adoptant une notion mentale de conceptualisation, nous prêtons le flanc aux critiques fameuses de Frege et Husserl concernant la variabilité des représentations mentales, sources de leur antipsychologisme notamment des significations. Nous abandonnons l'objectivité confortable des universaux et des objets en soi les instanciant pour leur substituer des idées privées subjectives avec pour charge de devoir expliquer comment différents sujets peuvent penser à la même chose. L'explication tient en deux thèses aujourd'hui très largement partagées en psychologie cognitive et philosophie du langage. D'une part, le concept est à distinguer de son contenu – la *conception* – pour rendre compte à la fois de l'endurance du concept et de la dynamicité de son contenu. D'autre part, la *similarité* de nos conceptions se rapportant à de mêmes objets crée une *intersubjectivité* permettant de rendre compte des conditions de communications réussies.

Le concept twardowskien partage avec la notion de *dossier mental* la propriété de véhiculer un contenu, dénommé « conception » du concept/dossier. Dans le cas du concept twardowskien, la conception correspond à un objet doté de propriétés. Selon François Recanati [28], cette distinction entre concept et conception est importante pour permettre d'expliquer qu'un sujet puisse « penser à la même chose » à différents moments, en y pensant éventuellement différemment. Un même concept continuant est mobilisé tandis que son contenu est susceptible d'évoluer au gré d'acquisitions de nouvelles informations.

Il reste à expliquer le « partage » des conceptions, autrement dit le fait que plusieurs sujets peuvent « penser à la même chose ». L'explication repose sur l'existence de différents processus psycho-sociaux. L'un de ces processus est le baptême d'entités particulières (personnes, animaux, lieux et objets variés) consistant à les doter d'un nom propre pour y faire référence, cette référence pouvant être transmise de générations en générations à l'occasion de communications – un tel processus est défini par Saul Kripke [17], à qui on attribue la paternité. Un autre de ces processus est l'usage d'actes de discours ostentatoires (ex : « Ceci est jaune », « ceci est un chien ») permettant à un sujet d'acquérir l'extension de concepts – ce processus est décrit par Hilary Putnam [27]. Nous pouvons également mentionner l'inculcation de théories scientifiques dans l'enseignement. Tous ces processus concourent à expliquer la similarité des conceptions de sujets participant d'une même communauté linguistique et culturelle. Recanati [29] élabore une notion idéale de *dossier mental partagé* pour expliquer l'identité des pensées et jugements, tout en reconnaissant que seuls existent des dossiers mentaux privés.

Nous venons d'admettre une dimension de construit social pour les conceptions des concepts, mais un constat s'impose : même si nous admettons une similarité des conceptions de nos concepts se référant à des objets donnés, tous les sujets ne partagent pas les mêmes concepts. Une question dès lors se pose : quels critères de choix des concepts (objets généraux) retenir pour établir une ontologie de référence qui facilite le partage des connaissances, ce qui correspond au rôle attribué en Ontologie appliquée aux ontologies fondatrices.

4 Ontologies descriptives *vs* révisionnistes

Aussi bien en Ontologie appliquée qu'en métaphysique se pose, à un premier niveau, la question de la place à accorder aux catégories de sens commun, scientifiques et celles proprement métaphysiques. En métaphysique, comme le souligne Jean-Baptiste Guillon [10], il est courant depuis Peter F. Strawson [34] de distinguer deux projets, à savoir une métaphysique *descriptive* et une métaphysique *révisionniste*. Strawson définit ainsi cette distinction : « La métaphysique descriptive se contente de décrire la structure actuelle de notre pensée sur le monde, tandis que la métaphysique révisionniste s'efforce de produire une meilleure structure » [*ibid.*, p. 9]. Strawson lui-même s'inscrit dans un projet de métaphysique descriptive, que l'on peut résumer ainsi : (i) la priorité est donnée à l'analyse de nos schèmes conceptuels se rapportant au monde plutôt qu'au monde directement ; (ii) les catégories utilisées par la pensée commune sont déjà bien adaptées à notre compréhension du monde et méritent d'être étudiées et clarifiées. A contrario, un projet révisionniste conserve (i), mais remet en cause (ii) en lui substituant, toujours selon Strawson, (ii)' : notre cadre conceptuel ordinaire pouvant être défectueux ou inadéquat, il faut s'efforcer de proposer de nouvelles structures et de nouveaux concepts qui reflèteraient mieux la nature du monde. Le point (i) n'exprime pas une thèse métaphysique mais un projet préconisant de prioriser l'étude de nos concepts par rapport à celle des entités mondaines en soi. Une priorité inverse est notoirement revendiquée par les auteurs de BFO, comme rappelé encore récemment [23] : « L'objectif d'une ontologie est de décrire la réalité. La recherche scientifique s'intéresse aux caractéristiques générales de la réalité et aux relations qui existent entre elles. Par conséquent, BFO consiste fondamentalement en une représentation de la réalité plutôt que du langage, des concepts ou des représentations mentales de la réalité ». Ce « principe de réalisme », ainsi nommé, même s'il n'est pas explicitement revendiqué par les auteurs des ontologies courantes, constitue un principe tacite. En témoigne le fait, rappelé en § 2, que les entités de discours des conceptualisations des ontologies référentielles sont des entités en soi instanciant des universaux.

Concernant le point (ii), on peut constater que, dans le domaine de l'Ontologie appliquée, le rôle dévolu au sens commun fait l'unanimité. Dans l'ontologie de la physique développée par Jobst Landgrebe et Smith [18], l'ontologie BFO est considérée remplir le rôle de composant de sens commun pour rendre compte des entités relevant de la physique classique. De leur côté, les auteurs de DOLCE ont systématiquement revendiqué la primauté du sens commun dans la sélection des catégories, mettant en avant un biais linguistique et cognitif, lequel [20] « repose sur l'hypothèse que la structure superficielle du langage naturel et ce que l'on appelle le sens commun ont une pertinence ontologique. Par conséquent, les catégories renvoient à des artefacts cognitifs qui dépendent plus ou moins

de la perception humaine, des empreintes culturelles et des conventions sociales. Dans le cadre de cette approche, il n'y a pas de restrictions majeures à la postulation de catégories ontologiques, car les paradigmes philosophiques ou scientifiques globaux sont négligés ». Pour notre part, si nous accordons au sens commun une valeur épistémique et aléthique, nous considérons qu'il doit être étayé par des connaissances scientifique ainsi que par des principes proprement ontologiques définissant des modes d'existence des entités en soi.

Ainsi, pour notre part, admettant (i), en guise de (ii) (et (ii)') nous mettons en avant le projet d'une métaphysique « descriptive étayée par la science et l'ontologie », définie comme suit :

- (iia) le sens commun doit être étayé et complété par des savoirs scientifiques apportant la norme du vrai ; le sens commun doit être confronté à nos connaissances scientifiques des entités en soi.
- (iib) une théorie ontologique est indispensable pour apporter une norme de l'existence ; distinguer différents modes d'existence est nécessaire pour distinguer le monde en soi (l'ontologie) et nos connaissances de ce monde (l'épistémologie).

En guise de premier exemple pour illustrer (iia-b), considérons celui des personnages de fiction. Avant toute chose, il convient de noter que lorsque l'on parle de « personnage de fiction », ce terme est homonymique car il désigne un *personnage*$_1$ créé par un auteur pour faire exister un *personnage*$_2$ dans un monde fictif. Ainsi, on doit distinguer *Sherlock Holmes*$_1$ créé par Conan Doyle, apparu en 1887 dans le roman *Une étude en rouge*, et *Sherlock Holmes*$_2$, ce détective privé vivant dans le Londres du 19ème siècle, violoniste aguerri, cocaïnomane et maître dans la résolution des énigmes. *Sherlock Holmes*$_1$ est un artefact culturel existant dans le monde réel, un objet mental possédant une identité sociale. Pour notre propos, nous nous intéressons particulièrement à *Sherlock Holmes*$_2$, une entité que l'ontologie populaire fait exister dans un « monde » créé par Conan Doyle.

Récemment, une étude conduite par Carola Barbero et coll. [2] a permis de valider psychologiquement cette notion de monde fictif. Ces auteurs ont conduit l'expérimentation suivante auprès d'une centaine de sujets profanes en matière de métaphysique. Les sujets devaient évaluer les conditions de vérité de phrases telle « Emma Bovary existe et Barack Obama existe », « Sherlock Holmes existe et Anna Karénine existe » ou « Pénélope Cruz existe et Snazzo existe », faisant varier les termes pour référer à des objets réels, fictifs et non existants. Selon Barbero et coll. [*ibid.*], la meilleure interprétation psychologique des résultats est que les sujets considèrent que : (i) les termes tel « Barack Obama » et « Emma Bovary » renvoient à des objets existants, au contraire de termes tel « Snazzo » ; toutefois, (ii) les objets réels et fictifs ne peuvent se rencontrer, vivant dans des mondes séparés, ce qui les empêche d'avoir des interactions causales. Ces résultats de psychologie cognitive viennent valider la thèse du *réalisme modal* défendue par David Lewis [19] conférant aux mondes réel et fictif ainsi qu'aux entités les peuplant une même nature.

En guise de second exemple, considérons l'analyse contemporaine interdisciplinaire (psychologie, physiologie, chimie, physique) des couleurs faisant référence menée par Alex Byrne et David R. Hilbert [4]. Une telle analyse conduit à distinguer, du côté mental, un percept phénoménal exprimable sous la forme d'une qualité inhérente à l'objet coloré dotée d'une magnitude et, du côté physique, une multitude de processus d'absorption et de réémission de la lumière à la surface de l'objet [24,25]. De fait, on peut considérer que nous disposons de deux conceptions de la couleur de l'objet matériel, l'une de sens commun – la couleur est une qualité *dans* (*inhérente à*) l'objet – l'autre scientifique – la couleur est un phénomène processuel affectant l'objet. Dans une ontologie épistémique, les deux conceptions sont considérées comme coréférentielles (il y a donc complémentarité et non révision). Chacune des conceptions joue un rôle cognitif spécifique. La seconde conception scientifique est à même d'expliquer la *constance* de la couleur de l'objet (dans des conditions standard d'illumination, l'objet nous apparaît comme ayant la même couleur). Pour une telle analyse, plusieurs théories ontologiques apportent leur concours (iib), notamment la théorie des *niveaux de la réalité* [1] permettant d'associer différentes qualités primitives composant la couleur (*teinte*, *brillance*, *saturation*) à différentes réalités physiques.

5 Approches multiplicatives *vs* réductionnistes de la réalité matérielle

Pour continuer à exposer la distinction entre ontologies référentielles et épistémiques, nous abordons la question de la nature des objets matériels, notamment des objets mésoscopiques – tables, chaises et autres objets du quotidien. En métaphysique, Peter van Inwagen [11] a fameusement avancé une thèse réductionniste réfutant l'existence de ces objets ordinaires, tandis que Kit Fine [5] a tout aussi fameusement avancé une thèse multiplicative défendant l'existence de *qua-objets* – littéralement des objets physiques « amalgamés » à une description conceptuelle. Un exemple de 'qua-objet' est celui d'une statue de Goliath identifiée à un bloc de bronze auquel un sculpteur a donné la forme de Goliath. Selon Fine, la statue et le bloc de bronze ayant des propriétés différentes (seule la statue a la forme de Goliath) sont deux objets matériels distincts. Une telle stratégie multiplicative a été reprise dans la communauté Ontologie appliquée par Laure Vieu et coll. [37] pour distinguer un artefact matériel et l'objet physique le constituant auquel une fonction a été attribuée (ex : un presse papier et un galet) ou un rôle et l'objet remplissant le rôle (ex : un individu passager d'un vol Air France). Fine a ultérieurement théorisé et étendu cette figure du 'qua-objet' avec les notions *d'incarnation rigide* et d'*incarnation variable* (*rigid et variable embodiment*) [6] pour rendre compte de la structure d'objets complexes composés d'autres objets. Afin d'effectuer un choix de théorie ontologique, plutôt que de s'en remettre a priori à un critère de nombre d'entités, par exemple à un principe de parcimonie, nous appliquons notre principe (iia) consistant à confronter ces théories ontologiques au sens commun étayé scientifiquement.

Les travaux de Elisabeth S. Spelke [33] en psychologie de la perception confèrent à ces objets du quotidien une réalité

mentale – nous les concevons comme des masses connectées de matière, solides, se mouvant comme un tout. De leur côté, Jean Petitot et Smith [26] ont soutenu que les sciences physiques sont à même de justifier l'existence de ces objets. Aussi bien la psychologie cognitive que les sciences physiques étayent l'existence respective de ces objets mentaux et matériels. Qu'en est-il toutefois de la nature des objets matériels en soi ?

Suivant la théorie de l'*incarnation* de Fine, un objet matériel complexe est un tout composé d'objets matériels et d'une entité intensionnelle rendant compte du principe de l'arrangement des composants en un tout. Par exemple, un sandwich au jambon est un tout composé de deux tranches de pain p_1 et p_2, d'une tranche de jambon j et du composant intensionnel *être entre* : $<p_1, p_2, j,$ *être entre*$>$ [6, p. 65-68]. Une telle théorie souffre à nos yeux d'un défaut important. Elle revient à introduire des objets inhomogènes sur un plan métaphysique – ayant pour composants des entités matérielles et conceptuelles – habituellement considérés, pour cette raison d'inhomogénéité, comme des chimères ontologiques. Nous formulons la même remarque à propos de la statue de Goliath dont le critère d'identité intègre le concept du personnage mythique de Goliath, un composant non matériel. Or, il existe une voie spécifiquement matérielle pour rendre compte d'objets complexes. Tout d'abord, le poids de la tranche supérieure de pain d'un sandwich tenu à l'horizontale est suffisant pour assurer son intégrité. Pour prévenir des mouvements pouvant mettre à mal cette intégrité, de la matière grasse (ex : du beurre) peut être ajoutée qui améliore l'adhésion des tranches de pain et du jambon. Enfin, si cela ne suffit pas, pour faciliter le transport du sandwich, un morceau de papier enveloppant le sandwich peut être ajouté. Bref, avec ou sans ajout de matière, différentes *connexions* physiques assurant l'intégrité du tout existent [15][2]. Le sandwich peut très bien être pensé comme composé d'une tranche de jambon se trouvant entre des tranches de pain ou pensé comme étant mon repas du midi mais, selon une ontologie épistémique, ces différents aspects sur un objet matériel correspondent à différents objets pensés représentant un même objet physique. La multiplication des entités se trouve du côté mental et non physique[3].

La motivation de la théorie de Fine est de palier les déficiences de la méréologie standard, laquelle conduit à considérer à la fois des parties et des fusions arbitraires d'objets en retenant comme seul critère l'extension spatiale des objets. Dès lors, la méréologie standard identifie un objet assemblé (un sandwich ou un vélo) au tas de ses parties désassemblées, les deux objets ayant les mêmes parties. Comme le précise Kathrin Koslicki [16, p. 171] « La méréologie standard donne la réponse très révisionniste selon laquelle pour chaque pluralité d'objets, aussi disparates et dissemblables soient-ils, le monde contient un autre objet ». Pour autant, doit-on la condamner comme radicalement révisionniste ?

Qu'il s'agisse de parties ou de fusions arbitraires, il y a bien dans le monde physique quelque chose de matériel existant, cependant ce quelque chose dépend d'un (ou plusieurs) objets ordinaires sans être lui-même un objet ordinaire. Smith [31] dans sa théorie néo-aristotélicienne de la substance (fondée sur l'existence d'une frontière externe) identifie ce quelque chose à une *entité substantielle* n'étant pas elle-même une substance. Deux catégories de telles entités existent, celles *bona fide* délimitées par des frontières physiques naturelles correspondant à des discontinuités qualitatives (ex : les parties du corps humain) et celles *fiat* délimitées uniquement par l'intellect (ex : les hémisphères d'une boule ronde parfaitement homogène[4]). Comment interpréter de telles analyses dans le cadre d'une ontologie épistémique ? Ces considérations de parties ou de fusions arbitraires d'objets (qu'on restreindra toutefois à des objets existants concomitamment) ne rajoutent rien à l'ontologie du monde physique. En revanche, qu'il s'agisse d'objets *fiat* ou *bona fide*, il convient de considérer que nous tenons autant d'objets pensés mentaux. À nouveau, la multiplication est du côté du mental et non du physique et si on ne se limite pas au physique, il n'y a pas lieu de considérer la méréologie standard comme révisionniste.

6 Conclusion

La notion d'ontologie épistémique se fonde sur une conception de la représentation conceptuelle admettant l'existence d'objets immanents à nos concepts. Cette thèse ontologique et psychologique, qui signe notamment la réhabilitation de la psychologie cognitive en métaphysique, entraîne plusieurs conséquences.

D'une part, elle invite à porter un regard nouveau sur nos pratiques de représentation des connaissances en considérant que les instances de nos bases de connaissances ne représentent pas directement des entités mondaines en soi mais représentent la façon dont ces entités sont pensées par des sujets. Ce point de vue soulève la question de l'objectivité de ces instances mais, comme nous l'avons défendu, ces objets mentaux privés sont des construits sociaux bien plus objectifs que n'ont voulu le faire croire des philosophes mathématiciens comme Frege et Husserl. Par ailleurs, cette thèse ontologique invite à maintenir une séparation stricte entre le matériel et le mental, évitant ainsi le recours à des entités hybrides. Ce principe conduit à des analyses différentes de celles couramment établies avec les ontologies référentielles. La notion d'*incarnation* proposée par Fine [6] pour rendre compte de la composition d'objets matériels, que nous avons critiquée, en est un exemple.

Dans ce texte, nous avons posé les premières pierres d'une ontologie épistémique fondatrice, de référence, un chantier auquel nous entendons nous consacrer à l'avenir.

Références

[1] L. Albertazzi and R. Poli, Multi-leveled objects: color as a case study, *Frontiers in Psychology*, Vol. 5, Article 592, 2014.

[2] C. Barbero, F. Domaneschi, I. Enrici and A. Voltolini, What is Existence? A Matter of Co(n)text, *Acta Analytica*, Vol. 39, pp. 1-39, 2024.

[2] Dans cette référence, nous nous appuyons sur les prémices d'une ontologie des *connexions* proposée par Frédéric Nef [22].

[3] Voir [13] pour une analyse détaillée.

[4] Smith a développé une théorie des objets *fiat* dans [30].

[3] S. Borgo, A. Galton and O. Kutz, Foundational ontologies in action, *Applied Ontology*, Vol. 17, No. 1, pp. 1-16, 2022.

[4] A. Byrne and D.R. Hilbert, Color realism and color science, *Behavioral and Brain Science*, Vol. 26, No 1, pp. 3-21, 2003.

[5] K. Fine, Acts, Events and Things, in *Language and Ontology*. Proc. of the *Sixth International Wittgenstein Symposium* (pp. 97-105), Vienna: Holder-Pichler-Tempsky, 1982.

[6] K. Fine, Things and Their Parts, *Midwest Studies in Philosophy*, Vol. 23, No 1, pp. 61-74, 1999.

[7] T.R. Gruber, A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, Vol. 5, No 2, pp. 199-220, 1993.

[8] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer Studies*, Vol. 43, No 5/6, pp. 907–28, 1995.

[9] N. Guarino and P. Giaretta, Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars (ed.), *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing* (pp. 25–32), Amsterdam: IOS Press, 1995.

[10] J.-B. Guillon, Le sens commun comme principe en métaphysique, dans A. Declos & J.-B. Guillon (dir.), *les principes métaphysiques*, Paris : Collège de France, 2020.

[11] P. van Inwagen, *Material Beings*, Ithaca, Cornell University Press, 1990.

[12] G. Kassel, A plea for epistemic ontologies, *Applied Ontology*, Vol. 18, No. 4, pp. 367-397, 2023.

[13] G. Kassel, Fondements ontologiques de la connaissance conceptuelle du monde matériel, dans T. Cazenave (ed.), actes de la *Conférence Nationale en Intelligence Artificielle CNIA @ PFIA 2025*.

[14] G. Kassel, Le legs twardowskien d'une ontologie épistémique, *Philosophiques*. *À paraître*.

[15] G. Kassel, Connexions et relations, *ROIA*. *À paraître*.

[16] K. Koslicki, *The Structure of Objects*. Oxford University Press, 2008.

[17] S.A. Kripke, *Naming and Necessity*. Cambridge, MA: Harvard University Press, 1980.

[18] J. Landgrebe and B. Smith, *Ontologies of common sense, physics and mathematics*, arXiv:2305.01560v1, 2023.

[19] D. Lewis, *De la pluralité des mondes*, Éditions de l'éclat : Paris / Tel Aviv, 2007 ; traduction par M. Caveribère et J.-P. Cometti de *On the Plurality of Worlds*, B. Blackwell, 1986.

[20] C. Masolo, S. Borgo, A. Gangemi, N. Guarino and A. Oltramari, The WonderWeb Library of Foundational Ontologies and the DOLCE ontology, WonderWeb Deliverable D18, Final Report, vr. 1.0, 2003.

[21] M. Murez and F. Recanati, Mental files: an Introduction, *Review of Philosophy and Psychology*, Vol. 7, No 2, pp. 265-281, 2016.

[22] F. Nef, *Les propriétés des choses. Expérience et logique*, Collection « Problèmes & Controverses », Paris, Vrin, 2006.

[23] J.N. Otte, J. Beverley and A. Ruttenberg, BFO: Basic Formal Ontology, *Applied Ontology*, Vol. 17, No 1, pp. 17-43, 2022.

[24] R. Pasnau, A Theory of Secondary Qualities, *Philosophy and Phenomenological Research*, Vol. 73, No 3, pp. 568-591, 2006.

[25] R. Pasnau, The event of color, *Philosophical Studies*, Vol. 142, No 3, pp. 353-369, 2009.

[26] J. Petitot and B. Smith, Physics and the Phenomenal World, in R. Poli & P.M. Simons (eds.), *Formal Ontology* (pp. 233-254), Nijhoff International Philosophy Series, Vol. 53, Dordrecht: Springer, 1996.

[27] H. Putnam, The Meaning of Meaning, in *H. Putnam, Mind, Language and Reality, Philosophical Papers* (pp. 215-271), Vol. 2, Cambridge: Cambridge University Press, 1975.

[28] F. Recanati, *Mental files*, Oxford University Press, 2012.

[29] F. Recanati, *Mental Files in Flux*, Oxford University Press, 2016.

[30] B. Smith, Fiat Objects, in N. Guarino, L. Vieu & S. Pribbenow (eds.), *Parts and Wholes: Conceptual Part-Whole Relations and Formal Mereology, proc. of the 11th European Conference on Artificial Intelligence*, pp. 15–22, 1994.

[31] B. Smith, Objects and Their Environments: From Aristotle to Ecological Ontology, in A.U. Frank, J. Raper & J.-P. Cheylan (eds.), *The Life and Motion of Socio-Economic Units* (pp. 79-97), London: Taylor & Francis, 2001.

[32] B. Smith, Ontology, in L. Floridi (ed.), *Blackwell Guide to the Philosophy of Computing and Information* (pp. 155-166), Oxford: Blackwell, 2003.

[33] E.S. Spelke, Principles of Object Perception, *Cognitive Science*, Vol. 14, pp. 29-56, 1990.

[34] P.F. Strawson, *Individuals. An Essay in Descriptive Metaphysics*, Routledge, 1959.

[35] K. Twardowski, Sur la théorie du contenu et de l'objet des représentations, dans J. English (ed.), *Husserl – Twardowski, sur les objets intentionnels (1893-1901)*, Paris, Vrin, pp. 85-200, 1993 ; traduction, introduction et notes, par J. English de *Zur Lehre vom Inhalt und Gegenstand der Vorstellungen. Eine psychologogische Untersuchung*, Vienne, Hölder, 1894.

[36] K. Twardowski, The Essence of Concepts, dans J. Brandl & J. Woleński (eds.), *Kazimierz Twardowski: On Actions, Products and other topics in philosophy* (pp. 73-98), Rodopi, 1999 ; traduction par A. Szylewicz de Über begriffliche Vorstellungen, Wissenschaftliche, Beilage zum 16. Jahresberichte der philosophischen Gesellschaft an der Universität zu Wien (pp. 1-28), Leipzig: Barth, 1903/1914.

[37] L. Vieu, S. Borgo and C. Masolo, Artefacts and Roles: Modelling Strategies in a Multiplicative Ontology, in C. Eschenbach & M. Grüninger (eds.), Proc. of the *fifth Internation Conference on Formal Ontology in Information Systems (FOIS 2008)* (pp. 121-134), IOS Press, 2008.

Vers un cadre ontologique pour la gestion des compétences : à des fins de formation, de recrutement, de métier, ou de recherches associées

Ngoc Luyen Le[1,2], Marie-Hélène Abel[2], Bertrand Laforge[1,3]

[1]Gamaizer, 93340 Le Raincy, France
[2]Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60319 - 60203 Compiègne Cedex, France
[3]Sorbonne Université, CNRS UMR 7585, LPMHE (Laboratoire de Physique Nucléaire et des Hautes Énergies), 75252 Paris cedex 05, France

Résumé

La transformation rapide du marché du travail, alimentée par les avancées technologiques et l'économie numérique, exige un développement continu des compétences et une adaptation constante. Dans ce contexte, les systèmes traditionnels de gestion des compétences manquent d'interopérabilité, d'adaptabilité et de compréhension sémantique, rendant difficile l'alignement des compétences individuelles avec les besoins du marché du travail et les formations. Cet article propose un cadre basé sur l'ontologie pour la gestion des compétences, permettant une représentation structurée à différents niveaux de granularité, allant des micro-capacités aux macro-compétences, afin de répondre à des besoins liés aux applications métiers ainsi qu'à la structuration des compétences dans les contextes professionnels et de formation. En exploitant des modèles ontologiques et un raisonnement sémantique, ce cadre vise à améliorer l'automatisation de l'appariement compétences-métiers, la personnalisation des recommandations d'apprentissage et la planification de carrière. Cette étude discute de la conception, de la mise en œuvre et des applications potentielles du cadre, s'appuyant sur la recherche de formation des compétences, la recherche d'un emploi et la recherche de personnes compétentes.

Mots-clés

Ontologies, Base de connaissances, Cadre ontologique, Gestion de compétences, Granularité de compétence.

Abstract

The rapid transformation of the labor market, driven by technological advancements and the digital economy, requires continuous competence development and constant adaptation. In this context, traditional competence management systems lack interoperability, adaptability, and semantic understanding, making it difficult to align individual competencies with labor market needs and training programs. This paper proposes an ontology-based framework for competence management, enabling a structured representation of competencies, occupations, and training programs. By leveraging ontological models and semantic reasoning, this framework aims to enhance the automation of competence-to-job matching, the personalization of learning recommendations, and career planning. This study discusses the design, implementation, and potential applications of the framework, focusing on competence training programs, job searching, and finding competent individuals.

Keywords

Ontology, Knowledge base, Ontological Framework, Competence Management, Granularity of Competence.

1 Introduction

La transformation rapide du marché du travail, portée par les avancées technologiques et la transition vers une économie numérique, impose une évolution constante des compétences professionnelles ainsi qu'une annotation plus fine et performante des compléments au diplôme. Les approches traditionnelles de gestion des compétences rencontrent plusieurs défis, notamment le manque d'interopérabilité, la rigidité des taxonomies et l'absence d'une représentation sémantique permettant un alignement efficace entre les compétences des individus, celles liées aux offres d'emploi et celles associées aux formations disponibles [3, 2]. Cette inadéquation complique la mise en place de stratégies de développement des compétences adaptées aux besoins du marché du travail. Au-delà des enjeux professionnels, cette difficulté existe également dans l'accompagnement pédagogique des étudiants tout au long de leur formation. Il s'agit notamment de mieux relier les compétences ou capacités spécifiques qu'ils mobilisent dans une activité pédagogique à l'ensemble des macro-compétences visées par la formation.

Face à ces défis, l'adoption d'une approche ontologique apparaît comme une solution innovante permettant de modéliser et de relier, de manière structurée, les connaissances relatives aux compétences, aux métiers et aux formations [1, 11]. En s'appuyant sur des standards reconnus (ROME[1], ESCO[2]) , il devient possible de développer un cadre pour la gestion des compétences. Ce cadre favorise l'interopérabilité entre différents systèmes (plateformes de formation, bases de données de métiers et d'offres d'emploi, etc.) et facilite la recommandation de formations, d'évolutions professionnelles ou de recrutement. En exploitant les capacités du web sémantique et des techniques de raisonnement automatisé, une ontologie permet de formaliser la représentation des compétences, des métiers et des formations [6, 11]. Cette approche vise à améliorer le rapprochement compétences-métiers, personnalise les recommandations de formation et anticipe mieux les évolutions

1. ROME : Répertoire Opérationnel des Métiers et des Emplois, https://www.francetravail.org
2. ESCO : European Skills, Competences, Qualifications and Occupations, https://ec.europa.eu/esco

des compétences requises.

Cet article présente un cadre ontologique pour la gestion des compétences, conçu pour répondre aux besoins actuels en matière de formation, de recrutement et d'évolution professionnelle. Ce cadre vise à assurer une meilleure correspondance entre les compétences des individus, les exigences du marché du travail, les opportunités d'emploi et les offres de formation. Basé sur une ontologie structurée, il modélise les compétences et établit leurs relations avec les métiers, tout en intégrant des mécanismes d'inférence sémantique et des algorithmes de recommandation. Ce dispositif permet d'identifier les formations les plus adaptées aux objectifs professionnels des utilisateurs, en facilitant leur montée en compétences et en renforçant leur employabilité.

Ce cadre s'inscrit dans les dynamiques actuelles de recherche sur la personnalisation des parcours éducatifs, l'orientation professionnelle assistée et la cartographie dynamique des compétences. Il fait notamment l'objet d'expérimentations dans des environnements tels que la plateforme de jeux éducatifs Ikigai.games [3], qui propose des scénarios interactifs visant à renforcer l'engagement des apprenants et le développement de compétences transversales dans des contextes variés.

La suite de cet article est organisée comme suit : la section suivante propose une revue des travaux connexes dans le domaine de la gestion des compétences et des ontologies appliquées. Elle est suivie d'une description détaillée de la problématique et du cadre ontologique proposé, incluant sa structure conceptuelle et les aspects liés à son implémentation. Nous présentons alors une étude de cas basée sur le référentiel ROME 4.0 afin d'illustrer notre approche. Enfin, l'article se conclut par une synthèse des principaux résultats obtenus et une discussion sur les perspectives futures de recherche et d'application.

Cadre	Couverture	Capacités Sémantiques	Cas d'Utilisation
ESCO Européen	Emplois, Compétences, Qualifications	RDF sans formalisation OWL	Planification de la main-d'œuvre, Éducation
O*NET États-Unis	Professions et Compétences professionnelles	Taxonomie de base	Appariement emploi, Planification de carrière
ROME France	Professions et Compétences professionnelles	RDF sans formalisation OWL	Services d'emploi
HR-XML Global	Échange de données RH	Aucun support sémantique	Intégration des systèmes RH
RNCP France	Certifications et qualifications professionnelles	Classification normalisée : référentiel reconnu officiellement	Validation des acquis, Formation professionnelle
SFIA Global	Compétences numériques et informatiques	Modèle structuré des compétences	Gestion des compétences IT, Développement professionnel
ISCO Global	Professions et compétences professionnelles	Classification hiérarchique : organisation des concepts en niveaux	Comparaison internationale, Statistiques sur l'emploi

TABLE 1. *Comparaison des Taxonomies et Standards de Compétences*

2 Travaux de la littérature

La gestion des compétences a été largement explorée dans les domaines de l'éducation, des ressources humaines et de l'IA. Deux approches dominent : les taxonomies classiques et les modèles ontologiques.

Les taxonomies (ESCO, ROME, O*NET, etc.) permettent de classifier les compétences et métiers. La Table 1 résume leurs portées, formats et usages. Bien qu'utiles pour la standardisation, ces modèles restent statiques et peu exploitables par des systèmes intelligents.

Pour répondre à ces limites, plusieurs travaux ont proposé des ontologies afin d'offrir une structuration dynamique, formelle et interopérable des compétences. Ces modèles permettent des inférences et recommandations automatisées. Par exemple, Miranda et al. [11] ont conçu une ontologie pour les systèmes RH, tandis que Paquette [12, 13] a modélisé les compétences en contexte éducatif. D'autres travaux [6, 14] s'intéressent à la personnalisation des parcours.

Cependant, peu de modèles relient explicitement compétences, métiers et formations, ou exploitent pleinement les règles d'inférence pour la recommandation. Notre travail vise à répondre à ces lacunes en proposant un cadre unifié, interopérable et orienté vers les cas d'usage formation/recrutement.

3 Problématique

Dans un contexte où la gestion des compétences doit s'adapter en permanence aux évolutions du marché du travail, plusieurs défis émergent. Pour un individu souhaitant exercer un métier, il est essentiel d'identifier les compétences requises et d'évaluer celles déjà acquises. Lorsqu'un écart de compétences est constaté, la question centrale est de déterminer quelles formations permettront d'acquérir les savoirs et savoir-faire nécessaires. Du point de vue des ressources humaines, le recrutement repose sur la capacité à évaluer les candidats en fonction des exigences d'un poste. Il s'agit de vérifier que la personne possède les compétences attendues ou, si nécessaire, de recommander des formations adaptées pour combler ces lacunes.

Un cadre ontologique structuré permettrait de modéliser et d'interconnecter les relations entre métiers, compétences et formations. Une telle approche faciliterait l'orientation professionnelle des individus, l'adéquation entre l'offre et la demande en compétences et l'optimisation des processus de recrutement. En exploitant les capacités des ontologies et du raisonnement sémantique, ce cadre contribuerait à une gestion plus efficace et dynamique des compétences, en anticipant les évolutions du marché, en améliorant la mobilité professionnelle et en renforçant l'adéquation entre les parcours éducatifs et les besoins du marché du travail. Dans la section suivante, nous présentons notre approche pour développer un cadre basé sur l'ontologie pour la gestion de compétences dans la prochaine section.

4 Cadre de gestion des compétences basé sur l'ontologie

Cette section décrit un cadre ontologique pour structurer les connaissances liées aux compétences, métiers et formations. Elle présente d'abord l'architecture en couches du système, puis détaille la modélisation ontologique des entités et leur intégration dans l'écosystème de gestion des compétences.

4.1 Achitecture du cadre ontologique

Le cadre repose sur une architecture en couches assurant la structuration des données, le raisonnement sémantique, les recommandations intelligentes, et l'interopérabilité avec les systèmes externes (Fig. 1).

3. https://ikigai.games/

Couche de données : regroupe compétences, métiers, formations et profils d'apprenants. Elle centralise les informations issues des référentiels et données contextuelles (comportementales, physiologiques) afin de structurer les ressources et personnaliser les parcours [15].

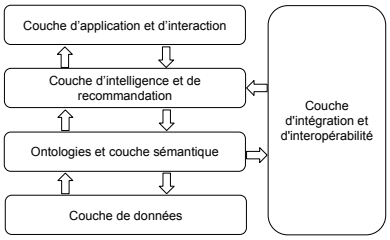

FIGURE 1. *Architecture du Cadre Ontologique pour la gestion de compétence*

Couche de sémantique : modélisation des entités (compétences, métiers, formations) en OWL/RDF, construction d'un graphe de connaissances aligné sur les référentiels. Inférences logiques (écarts, correspondances) utilisées par la couche suivante pour la recommandation [9].

Couche d'intelligence et de recommandation : moteur de recommandation basé sur les écarts de compétences, appariement profil-métier via IA, prédiction de trajectoires et suivi dynamique des compétences [4, 8, 5].

Couche d'application et d'interaction : interfaces pour apprenants (visualisation des compétences, recommandations), recruteurs (recherche de profils), et formateurs (gestion de l'offre pédagogique).

Couche d'intégration et d'interopérabilité : connexion aux plateformes LMS (Moodle, Coursera), outils de gestion de resources(Memorae), jeux éducatif (Ikigai.games), et systèmes d'information d'entreprise. Données personnelles conformes RGPD (pseudonymisation, consentement, traçabilité).

4.2 Ontologie de gestion des compétences

Le développement d'une ontologie pour la gestion des compétences permet de structurer et de formaliser les relations entre les compétences, les formations et les métiers. Cette section présente l'ontologie CMO - **C**ompetency **M**anagement **O**ntology, conçue pour servir de support à la gestion des compétences et à leur intégration dans un écosystème global de gestion des talents.

L'ontologie CMO repose principalement sur une structuration détaillée des compétences, prenant en compte leur classification ainsi que leur contexte métier. Comme illustré dans la figure 2, les *compétences* sont organisées en différentes catégories : compétences sociales (ex. Leadership, Communication), compétences cognitives (ex. Pensée critique, Raisonnement), compétences techniques (ex. Gestion de projet, Ingénierie) et compétences linguistiques (ex. Langues, Expression) [1, 7]. Chaque compétence peut être décomposée en sous-compétences, permettant ainsi une granularité plus fine dans l'évaluation et la structuration des connaissances et savoir-faire. En complément, l'ontologie intègre les *référentiels de compétences*, qui servent à normaliser la description et l'évaluation des compétences selon des standards établis. Elle distingue notamment les référentiels de compétence nationaux, définis par des institutions gouvernementales ou académiques, et les référentiels de compétence internationaux, alignés sur des modèles tels qu'ESCO, ROME, O*NET ou HR-XML. L'association des compétences à ces référentiels permet une meilleure compatibilité avec les cadres normatifs existants et facilite la reconnaissance des compétences au niveau international.

Les compétences sont directement liées aux métiers, permettant d'identifier les aptitudes nécessaires pour exercer une profession. Chaque métier est caractérisé par son secteur d'activité, qui définit son domaine économique, ainsi que par son contexte de travail, précisant les conditions et les exigences spécifiques associées à l'emploi. L'ontologie prend également en compte la mobilité professionnelle, en modélisant les transitions possibles entre métiers en fonction des compétences transférables. De plus, elle intègre des notions d'enjeux et de thèmes clés, mettant en lumière les tendances et évolutions qui influencent l'évolution des compétences dans un domaine donné.

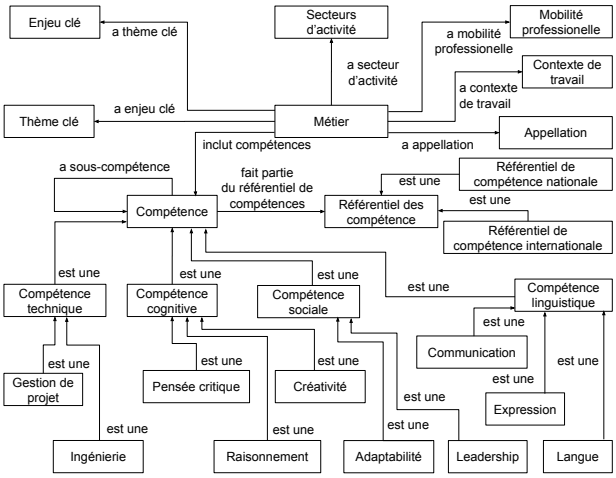

FIGURE 2. *Modélisation ontologique des compétences et de leur contexte d'application*

Sur le plan structurel, l'ontologie repose sur des relations sémantiques qui définissent les interconnexions entre les entités. Parmi celles-ci, la relation "*a sous-compétence*" permet de détailler une compétence en sous-éléments et facilitant ainsi la représentation de compétences composites ou hiérarchisées, tandis que "*fait partie du référentiel de compétences*" associe une compétence à un référentiel de compétences spécifique. La relation "*a secteur d'activité*" établit le lien entre un métier et son domaine professionnel, et "*a enjeu clé*" et "*a thème clé*" identifient les défis et évolutions qui impactent les besoins en compétences d'un métier. Enfin, "*a contexte travail*" décrit l'environnement dans lequel une compétence est mise en pratique.

À cette structuration des compétences vient se positionner les profils des individus/apprenants, les formations ou cours suivis, et les certifications obtenues, et les expériences professionelles. Les entités temporelles et des niveaux de compétences viennent alors compléter le modèle de façon à permettre d'évaluer dynamiquement les profils, comme illustré dans la figure 3. En particuler, chaque individu/apprenant possède un profil d'individu/apprenant, qui regroupe l'ensemble de ses compétences acquises soit par formation, soit par expérience professionnelle. Chaque compétence est associée à un niveau de compétence, permettant de quantifier l'expertise acquise par un individu/apprenant. Ce niveau peut être précisé à l'aide de scores et d'étiquetages sémantiques ("*a niveau compétence*", "*a score max*", "*a score*"), offrant ainsi une granularité fine dans l'évaluation des compétences.

Les formations jouent un rôle clé dans le développement

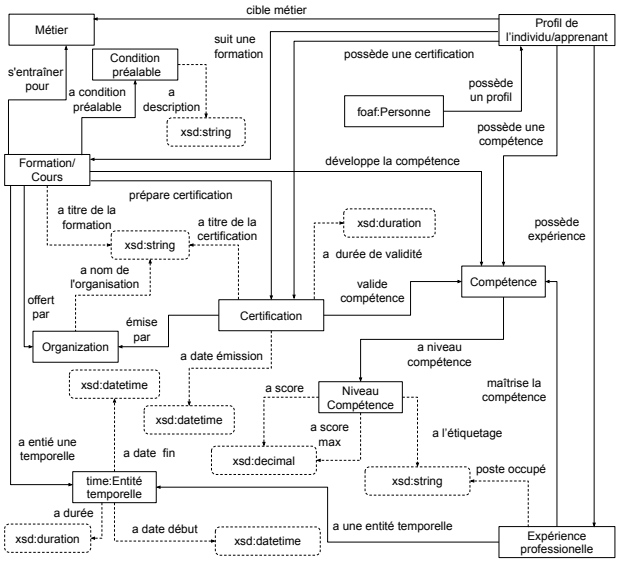

FIGURE 3. *Intégration des compétences avec les formations et les certifications*

des compétences. Un individu/apprenant peut suivre une formation dispensée par une organisation, celle-ci pouvant être une université, un centre de formation ou une entreprise. L'ontologie permet également de modéliser les conditions préalables ("*a condition préalable*"), garantissant que l'individu dispose des préconditions minimales requises avant d'accéder à une formation donnée. En complément des formations, les certifications viennent valider officiellement certaines compétences. Une certification est émise par une organisation et est directement liée à une compétence validée ("*valide compétence*"). Chaque certification possède une date d'émission ("*a date émission*") et une durée de validité ("*a durée de validité*"), assurant ainsi une gestion efficace des qualifications professionnelles.

Un élément central de cette ontologie est le processus de validation des compétences. Lorsqu'un individu/apprenant complète une formation, il peut obtenir une certification reconnue, garantissant ainsi la légitimité de ses acquis auprès des employeurs et des institutions. Cette certification est délivrée par une organisation accréditée et permet de formaliser la montée en compétence de l'individu. L'acquisition des compétences ne se limite pas aux formations ; elle peut également être consolidée à travers l'expérience professionnelle. Un individu possède une expérience professionnelle, qui est associée à un *poste occupé*. Cette expérience permet de maîtriser des compétences spécifiques et constitue un facteur clé dans l'évaluation des aptitudes professionnelles.

L'intégration d'une entité temporelle est un aspect fondamental de cette ontologie. Chaque formation, certification et expérience professionnelle est associée à une durée ("*xsd :duration*"), une date de début ("*a date début*") et une date de fin ("*a date fin*"). Cette structuration temporelle permet de suivre l'évolution des compétences d'un individu sur le long terme et d'optimiser la gestion des carrières. Afin de s'adapter à l'évolution rapide des métiers et des compétences, l'ontologie prévoit un processus de maintenance continue, reposant sur l'alignement régulier avec des référentiels actualisés (tels que ROME ou ESCO) et sur une validation collaborative par des experts métiers. Une interface d'édition permet en outre de proposer, réviser et valider de nouvelles compétences au fil du temps. Pour illustrer concrètement l'application de l'ontologie CMO et du cadre associé, une étude de cas détaillée est présentée dans la section suivante.

5 Étude de cas

Dans cette section, nous nous concentrons sur l'expérimentation du cadre ontologique et de l'ontologie CMO à travers une étude de cas visant à requalifier des individus/apprenants vers des métiers en forte demande selon le référentiel compétences ROME 4.0. Nous présentons le contexte et les objectifs, détaillons le scénario de l'étude de cas, explorons l'interrogation ontologique à l'aide de requêtes SPARQL, et discutons des résultats attendus et de leur impact.

5.1 Contexte et objectifs

Dans un marché du travail en constante évolution, accéléré par l'intelligence artificielle générative [10], de nombreux individus/apprenants cherchent à améliorer leurs compétences pour accéder à des métiers en forte demande. Ces évolutions sont particulièrement notables dans le cadre du ROME 4.0, qui propose une classification des métiers en fonction des compétences requises, permettant ainsi une meilleure correspondance entre l'offre et la demande. Dans ce contexte, nous proposons une approche ontologique permettant de modéliser les compétences, les parcours de formation et les exigences des métiers. Ce cadre formel facilite l'identification des écarts entre les compétences détenues par un individu et celles exigées par un métier cible. Grâce à cette approche, il devient possible de recommander des formations personnalisées, d'intégrer des systèmes de certification et d'assurer une interopérabilité avec des plateformes de gestion des talents et d'apprentissage en ligne.

5.2 Scénario de l'étude de cas

Dans ce scénario, nous considérons *Louis Le*, un apprenant souhaitant évoluer vers le métier de *Data Scientist* (Code du *ROME M1405*[4]). Actuellement, il possède des compétences en Python, mais uniquement à un niveau basique (*cmo :Niveau01*). Cependant, le métier de *Data Scientist* requiert des compétences avancées, notamment en *Python Avancé*, en *Machine Learning* et en *Analyse de Données*, que *Louis Le* ne maîtrise pas encore.

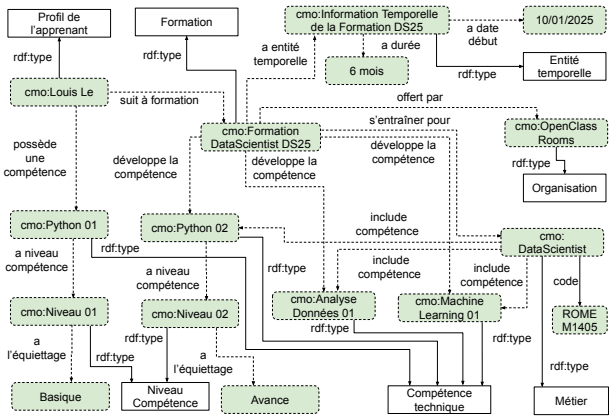

FIGURE 4. *Instance ontologique illustrant la relations d'un individu/apprenant vers un métier du ROME 4.0 en reliant ses compétences, et la formation recommandée.*

4. https://candidat.francetravail.fr/metierscope/fiche-metier/M1405/data-scientist

L'ontologie permet d'analyser son profil actuel et de le comparer aux exigences du métier ciblé, identifiant ainsi un écart de compétences. Afin de combler cet écart, une formation adaptée est recommandée : la "*Formation Data Scientist DS25*, proposée par *OpenClassrooms*. Comme illustré dans la Figure 4, cette formation est spécifiquement conçue pour permettre à *Louis Le* d'acquérir les compétences manquantes.

Une fois inscrit à la "*Formation Data Scientist DS25*", *Louis Le* pourra progressivement améliorer son niveau en *Python*, développer son expertise en *Analyse de Données* et acquérir de nouvelles compétences en *Machine Learning*. La formation, d'une durée de *6 mois* et commencée le *10/01/2025*, est structurée en plusieurs modules et est associée à un système d'évaluation des compétences.

À l'issue du programme, *Louis Le* devra valider ses acquis grâce à une évaluation formelle (*cmo :Niveau02* pour un niveau avancé en Python, par exemple). Si l'évaluation est réussie, une *certification* lui sera délivrée, attestant de son niveau en Data Science et facilitant son insertion professionnelle. L'intégration de cette ontologie avec des plateformes de certification et des systèmes RH permettra d'automatiser la mise à jour de son profil professionnel, garantissant ainsi une meilleure employabilité et une meilleure adéquation avec les offres du marché.

5.3 Requêtes SPARQL sur l'ontologie

L'utilisation des requêtes SPARQL permet d'extraire des informations clés sur les compétences d'un individu, les formations disponibles et les écarts à combler. Afin d'illustrer cette approche, nous proposons des requêtes visant à récupérer l'ensemble des compétences possédées par un individu, ainsi que leur niveau associé, et à identifier les compétences manquantes par rapport aux exigences du métier visé.

Dans la première requête SPARQL, illustrée dans la boîte de requête 1, l'objectif est de récupérer l'ensemble des compétences possédées par un individu, ainsi que leur niveau associé, si cette information est disponible.

Plus précisément, la requête interroge l'ontologie en exploitant le prédicat *possède une compétence – cmo :possedeCompetence*, qui établit une relation entre un individu (*Profil de l'apprenant – cmo :ProfilApprenant*) et les compétences qu'il a acquises. Par ailleurs, une jointure optionnelle est effectuée à l'aide de la clause OPTIONAL, permettant de récupérer, lorsque disponible, le niveau de maîtrise correspondant à chaque compétence (*a niveau compétence – cmo :aNiveauCompetence*). Cette approche garantit que les individus dépourvus d'une information explicite sur leur niveau de compétence ne soient pas exclus des résultats.

Listing 1. *Requête SPARQL pour récupérer les compétences et niveaux associés d'un individu*

```
PREFIX cmo:  <http://gamaizer.ia/cmo#>
PREFIX rdf: <http://w3c.org/1999/02/22-rdf-syntax-ns#>
SELECT ?pa ?competence ?niveau
WHERE {
 ?pa rdf:type cmo:ProfilApprenant;
 cmo:possedeCompetence ?competence .
 OPTIONAL {
  ?competence cmo:aNiveauCompetence
    ?niveau .
 }
}
```

La Table 2 présente les résultats de le requête SPARQL 1 extrayant les compétences et niveaux associés à différents individus. Chaque ligne associe un individu à une compétence et à son niveau de maîtrise, lorsqu'il est renseigné.

TABLE 2. *Résultats de la requête SPARQL : Extraction des compétences et niveaux pour différents individus*

Individu	Compétence	Niveau
Louis Le	Python 01	Basique
Henri Le	Python 01	Avancé
Henri Le	Machine Learning 02	Intermédiaire
Henri Le	Big Data	*Non défini*
Marc	Cybersécurité	Expert
Marc	Réseau Informatique	Intermédiaire
Marc	Python 01	Débutant
Sophie	UX/UI Design	Avancé
Sophie	Développement Web	Intermédiaire
Sophie	Python	*Non défini*

Les résultats mettent en évidence des compétences sans niveau défini (*Non défini*), indiquant des acquisitions non évaluées ou en attente de validation. Cette information est essentielle pour identifier les écarts de compétences et recommander des formations adaptées. Par exemple, *Sophie*, experte en *UX/UI Design*, pourrait renforcer son profil en complétant sa maîtrise de *Python*, tandis que *Marc*, débutant en *Python*, pourrait bénéficier d'une montée en compétences ciblée.

Nous examinons une deuxième requête SPARQL, présentée dans la boîte de requête 2, dont l'objectif est d'identifier les compétences manquantes d'un individu souhaitant accéder à un métier cible. Plus précisément, cette requête compare les compétences requises pour le métier *M1405* (*Data Scientist*) avec celles déjà acquises par *Louis Le*, en retournant uniquement celles qu'il ne possède pas encore.

Listing 2. *Requête SPARQL pour identifier les compétences manquantes pour un métier cible*

```
PREFIX cmo: <http://gamaizer.ia/cmo#>
SELECT ?competenceRequise
WHERE {
 cmo:M1405 cmo:includeCompetence ?competenceRequise .
 FILTER NOT EXISTS {
  cmo:LouisLe cmo:possedeUneCompetence ?
       competenceRequise .
 }
}
```

La requête exploite la propriété *include compétence – cmo :includeCompetence*, qui associe un métier aux compétences nécessaires à son exercice. La clause FILTER NOT EXISTS permet d'exclure les compétences déjà détenues par *Louis Le*, identifiées via la propriété *possède une compétence – cmo :possedeUneCompetence*. Ainsi, seules les compétences exigées par le métier et absentes du profil de l'apprenant seront affichées dans les résultats.

TABLE 3. *Résultats de la requête SPARQL : Identification des compétences manquantes pour le métier M1405*

Individu	Compétence Manquante
Louis Le	Machine Learning 01
Louis Le	Analyse de données 01
Louis Le	Python_02

La Table 3 présente les résultats de la requête SPARQL visant à identifier les compétences manquantes de *Louis Le* pour accéder au métier *M1405* (*Data Scientist*). L'analyse des écarts de compétences révèle que *Louis Le* ne possède pas encore les compétences clés suivantes : *Machine Learning 01*, *Analyse de données 01* et *Python 02*.

Ces résultats indiquent que, bien que *Louis Le* ait déjà certaines compétences en programmation, il doit encore acquérir des connaissances avancées en Python (*Python 02*), ainsi qu'une maîtrise des techniques d'*analyse de données* et de *machine learning*. Ces compétences étant essentielles pour le poste visé, leur absence constitue un frein à son évolution vers le métier de *Data Scientist*.

Grâce à cette détection automatique, il est possible d'orien-

ter *Louis Le* vers des formations spécifiques qui combleront ses lacunes. Par exemple, une formation avancée en *Python* pour la *Data Science*, un cours en *Machine Learning* et une formation en *Analyse de Données* seraient des recommandations pertinentes pour renforcer son profil. Cette approche permet d'éviter des apprentissages redondants, de raccourcir le parcours de requalification et d'accélérer son intégration dans le marché du travail.

Les deux requêtes SPARQL présentées permettent d'analyser le profil de compétences d'un individu en identifiant à la fois les compétences acquises et celles manquantes pour un métier cible. La première requête extrait les compétences possédées par un apprenant ainsi que leur niveau de maîtrise, offrant ainsi un état des lieux précis de ses acquis. La seconde requête, quant à elle, compare ces compétences aux exigences du métier *M1405* (*Data Scientist*) et détecte automatiquement les écarts de compétences que l'apprenant doit encore acquérir.

5.4 Résultats attendus et impact

L'intégration de l'ontologie CMO vise à améliorer la personnalisation des parcours d'apprentissage et de requalification. En identifiant automatiquement les écarts entre compétences détenues et compétences requises, elle permet de recommander des formations ciblées, réduisant les redondances et optimisant le temps d'apprentissage.

Pour les apprenants, cela se traduit par un accompagnement individualisé et une meilleure employabilité. Pour les recruteurs, le système facilite l'identification de profils pertinents via une visualisation structurée des compétences, enrichie par l'analyse de niveaux, scores et expériences. Les fournisseurs de formation peuvent, de leur côté, adapter leur offre en fonction des besoins réels du marché.

L'interopérabilité avec des plateformes comme Moodle, LinkedIn, Memorae ou Ikigai.games garantit une synchronisation fluide des données, assurant une mise à jour continue des profils. Le système respecte par ailleurs les exigences du RGPD grâce à des mécanismes intégrés de consentement, pseudonymisation et traçabilité.

Enfin, le cadre permet un suivi dynamique des compétences : les recommandations évoluent avec le marché grâce à l'analyse sémantique et aux contributions d'experts métiers. Il constitue ainsi un outil stratégique pour la gestion des talents, la mobilité professionnelle et l'anticipation des besoins futurs en compétences.

6 Conclusion et perspectives

Dans cet article, nous avons conçu un cadre ontologique pour la gestion des compétences, à des fins de formation, de recruitement, ou de métier, afin d'optimiser l'alignement entre les exigences du marché du travail et les formations disponibles. En intégrant des mécanismes de recherche et d'inférence basés sur l'ontologie, cette approche permet d'identifier les écarts de compétences et de recommander des formations adaptées, améliorant ainsi l'employabilité et accélérant la requalification professionnelle. Ce cadre a fait l'objet d'une première validation exploratoire, posant les bases du développement d'un prototype et démontrant son potentiel pour structurer et automatiser la gestion des compétences. Plusieurs axes d'amélioration restent à explorer. L'intégration de modèles d'apprentissage automatique pourrait affiner les recommandations, en analysant plus finement l'évolution des compétences et les tendances du marché. Par ailleurs, une validation à plus grande échelle, associée à une meilleure interopérabilité avec les plateformes de formation et de recrutement, permettrait d'élargir l'impact du cadre et de renforcer son adoption. Des expérimentations futures sont ainsi envisagées pour évaluer sa robustesse, sa précision en contexte réel, ainsi que son acceptabilité par les utilisateurs. En combinant ontologies, inférence sémantique et intelligence artificielle, ce cadre évolutif soutient l'adaptation continue des compétences et facilite les transitions professionnelles dans un marché en mutation.

Remerciements

Nous remercions chaleureusement le consortium Ikigai porté par l'association Games for Citizens, la société Gamaizer ainsi que le projet FORTEIM (projet lauréat AMI CMA France 2030), pour leur soutien et leur collaboration. Leurs contributions ont apporté une valeur ajoutée significative à la réalisation de cette recherche.

Références

[1] Marie-Hélène Abel. *Apport des Mémoires Organisationnelles dans un contexte d'apprentissage*. PhD thesis, Université de Technologie de Compiègne, 2007.

[2] David Arribas-Aguila, Gloria Castaño, and Rosario Martínez-Arias. A systematic review of evidence-based general competency models : Development of a general competencies taxonomy. *Journal of Work and Organizational Psychology*, 40(2) :61–76, 2024.

[3] C Brittain and F Bernotavicz. Competency-based workforce development : A synthesis of current approaches. *National Child Welfare Workforce Institute, University at Albany, Albany, NY*, 2015.

[4] Nikolas Dawson, Mary-Anne Williams, and Marian-Andrei Rizoiu. Skill-driven recommendations for job transition pathways. *Plos one*, 16(8) :e0254722, 2021.

[5] Jens-Joris Decorte, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. Career path prediction using resume representation learning and skill-based matching. *arXiv preprint arXiv :2310.15636*, 2023.

[6] Fotis Draganidis, Paraskevi Chamopoulou, and Gregoris Mentzas. An ontology based tool for competency management and learning paths. In *6th International Conference on Knowledge Management (I-KNOW 06)*, pages 1–10, 2006.

[7] Charles Emmanuel Foveau. *Référentiels des compétences et des métiers : une approche ontologique*. PhD thesis, Chambéry, 2007.

[8] Ngoc Luyen Le, Marie-Hélène Abel, and Philippe Gouspillou. A constraint-based recommender system via rdf knowledge graphs. In *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 849–854. IEEE, 2023.

[9] Ngoc Luyen Le, Marie-Hélène Abel, and Philippe Gouspillou. Improving semantic similarity measure within a recommender system based-on rdf graphs. In *International Conference on Information Technology & Systems*, pages 463–474. Springer, 2023.

[10] Frédéric Marty. L'intelligence artificielle générative et actifs concurrentiels critiques : discussion de l'essentialité des données. Technical report, Groupe de REcherche en Droit, Economie, Gestion (GREDEG CNRS), Université Côte d'Azur, France, 2024.

[11] Sergio Miranda, Francesco Orciuoli, Vincenzo Loia, and Demetrios Sampson. An ontology-based model for competence management. *Data & Knowledge Engineering*, 107 :51–66, 2017.

[12] Gilbert Paquette. An ontology and a software framework for competency modeling and management. *Journal of Educational Technology & Society*, 10(3) :1–21, 2007.

[13] Gilbert Paquette, Olga Marino, and Rim Bejaoui. A new competency ontology for learning environments personalization. *Smart Learning Environments*, 8(1) :16, 2021.

[14] Kalthoum Rezgui, Hédia Mhiri, and Khaled Ghédira. An ontology-based approach to competency modeling and management in learning networks. In *Agent and Multi-Agent Systems : Technologies and Applications : Proceedings of the 8th International Conference KES-AMSTA 2014 Chania, Greece, June 2014*. Springer, 2014.

[15] Yevgeniya Sulema, Andreas Pester, Bertrand Laforge, and Frederic Andres. Augmented reality user's experience : Ai-based data collection, processing and analysis. In *Augmented Reality and Artificial Intelligence : The Fusion of Advanced Technologies*, pages 31–46. Springer, 2023.

Session 6 : Ingénierie des connaissances pour les humanités numériques

PeGazUs : une méthode de reconstitution de l'évolution des entités géographiques à partir de données hétérogènes et fragmentaires

Charly Bernard[1], Nathalie Abadie[1], Bertrand Duménieu[2], Julien Perret[1]

[1] LASTIG, Université Gustave Eiffel, IGN/ENSG
[2] CRH, EHESS-CNRS

charly.bernard@ensg.eu ; nathalie-f.abadie@ensg.eu ; bertrand.dumenieu@ehess.fr ; julien.perret@ensg.eu

Résumé

Constituer un référentiel géo-historique d'entités spatiales permet de nombreux cas d'application, comme l'étude des dynamiques urbaines. Différentes approches dans la littérature montrent qu'il est possible de construire un tel référentiel, mais imposent d'avoir des jeux de données homogènes et structurés, à différentes dates. La disponibilité croissante des sources d'archives numérisées et les progrès des méthodes d'extraction d'informations dans ces sources permettent désormais de produire de grands volumes de données hétérogènes, fragmentaires et incomplètes, sur les entités géographiques du passé et leurs évolutions. Or, les approches de création de référentiels géo-historiques existantes ne permettent pas encore d'intégrer ces types de données de façon satisfaisante. Dans cet article, nous proposons une méthode de reconstitution de l'évolution spatio-temporelle des entités géographiques à partir de données hétérogènes et fragmentaires provenant de différentes sources. Nous expliquons aussi la manière dont on vérifie la cohérence du graphe de données créé à partir de cette méthode. Enfin, nous mettons cette dernière en application sur le quartier de la Butte aux Cailles à Paris à partir de sources de la fin du XVIII^{e} siècle à nos jours.

Mots-clés

Index géographique urbain historique - Graphe de connaissances - Évolution des entités géographiques.

Abstract

Building a geo-historical repository of spatial entities can be used in a number of applications, such as the study of urban dynamics. Various approaches in the literature show that it is possible to build such a repository, but require homogeneous and structured datasets at different dates. The increasing availability of digitised archive sources and advances in methods for extracting information from these sources now make it possible to produce large volumes of heterogeneous, fragmentary and incomplete data on past geographical entities and their evolution. However, existing approaches to creating geohistorical reference systems are not yet able to integrate these types of data satisfactorily. In this article, we propose a method for reconstructing the spatio-temporal evolution of geographical features using heterogeneous and fragmentary data from different sources. We also explain how to check the consistency of the data graph created using this method. Finally, we apply the method to the Butte aux Cailles district of Paris, using sources from the late 18th century to the present day.

Keywords

Historical urban gazetteer - Knowledge graph - Geographical entities evolution.

1 Introduction

Un index géographique (ou *gazetteer*) est une liste de lieux dans laquelle on représente, pour chaque lieu, un nom, un type et lorsque c'est possible, une localisation la plupart du temps sous la forme de coordonnées géographiques [18]. Représenter ce type d'informations à l'échelle des adresses présente un double intérêt. D'une part, cela permet d'indexer spatialement des documents d'archives numérisés, dont beaucoup regorgent de mentions d'adresses : c'est le cas des annuaires, des plans de ville anciens, des registres administratifs, ou encore des documents notariés par exemple. D'autre part, cela permet éventuellement de dater ces documents en fonction des adresses qu'ils mentionnent et de leurs dates d'existence connues. Au cours de la dernière décennie, un consensus s'est formé autour de l'utilisation de graphes de connaissances pour représenter des *gazetteers* historiques. Ceux-ci s'avèrent en effet particulièrement adaptés pour intégrer des données très hétérogènes et de structure non connue a priori [4].

Représenter des adresses anciennes dans un graphe de connaissances géo-historique nécessite une ontologie adaptée et une approche de peuplement capable d'intégrer des données issues de sources hétérogènes, à différentes dates et fragmentaires. En effet, bien que très représentées dans les sources historiques, les adresses sont des entités géographiques dont la généalogie est peu documentée. Ainsi, on ne dispose généralement que d'attestations d'existence, discontinues dans le temps, présentant tantôt des recouvrements temporels tantôt des lacunes.

Dans cet article, nous proposons une nouvelle approche pour peupler une ontologie représentant des adresses an-

ciennes et leur évolution au cours du temps. À partir de données décrivant des états ou des événements et provenant de différentes sources, avec différents temps valides, nous proposons de reconstituer pour chaque adresse ou élément de la voirie urbaine (rue, place, etc.) l'évolution de ses propriétés avec un enchaînement de versions successives reliées entre elles par des changements. La principale contribution de cette approche est de fournir une représentation continue et cohérente de l'évolution de chaque adresse, inférée à partir d'attestations d'états ou d'événements ponctuelles ou discontinues. L'approche utilise l'ontologie PeGazUs introduite par Bernard *et al.* [7].

Cet article est organisé de la façon suivante : nous présentons tout d'abord un état de l'art sur les méthodes de peuplement d'ontologies visant à reconstituer des évolutions spatio-temporelles du territoire. Ensuite, nous proposons une approche de reconstitution automatique de l'évolution temporelle d'adresses anciennes à partir de données fragmentaires pouvant représenter pour chaque entité géographique, son état sur un intervalle défini ou un événement décrivant son évolution à un instant donné. Puis, nous montrons que notre méthode fournit des données cohérentes et continues dans le temps au sein du graphe. Enfin, nous appliquons notre méthode sur les adresses du quartier de la Butte aux Cailles à Paris, au cours du XIX^{e} siècle.

2 État de l'art et notions préliminaires

2.1 Représentation des dynamiques spatio-temporelles

L'intégration de l'aspect temporel pour représenter les dynamiques territoriales a fait l'objet de travaux très tôt dans le domaine des systèmes d'information géographique [8, 17]. De nombreux travaux se sont concentrés sur la définition de l'identité des entités géographiques [19, 12, 16, 14]. Del Mondo [14] indique que la notion d'identité est cruciale pour conceptualiser et modéliser des phénomènes et qu'elle permet d'en transcrire une représentation dans une base de données. Hallot [19] reprend le modèle de l'*Identity Base Change* [20] et considère que chaque entité a une durée de validité infinie dont l'évolution est marquée par l'alternance d'états spatio-temporels (existence, non existence, etc.). Les travaux visant à décrire les dynamiques territoriales dans des SIG se sont largement fondés sur des modèles de graphes spatio-temporels [14, 15, 13].

Kauppinen *et al.* [21] proposent une ontologie pour représenter les municipalité finlandaises et y ntroduisent le concept de Change Bridge qui consiste à décrire les changements qui relient deux états successifs d'une entité géographique.

Bernard *et al.* [6] proposent une ontologie appelée TSN (Territorial Statistical Nomenclature) qui permet de représenter les unités territoriales de la NUTS [1]. Son extension TSN-Change permet de tracer l'évolution des entités géographiques grâce aux différentes versions de la NUTS en réutilisant le concept de Change Bridge. Alors que Kauppinen *et al.* [21] utilisent un changement pour décrire une transition entre deux états successifs, un changement pour [6] constitue un ensemble de modifications élémentaires sur plusieurs données. Par exemple, lors de la fusion deux entités impliquant la création d'une troisième, un changement va modéliser trois modifications élémentaires : deux disparitions et une apparition.

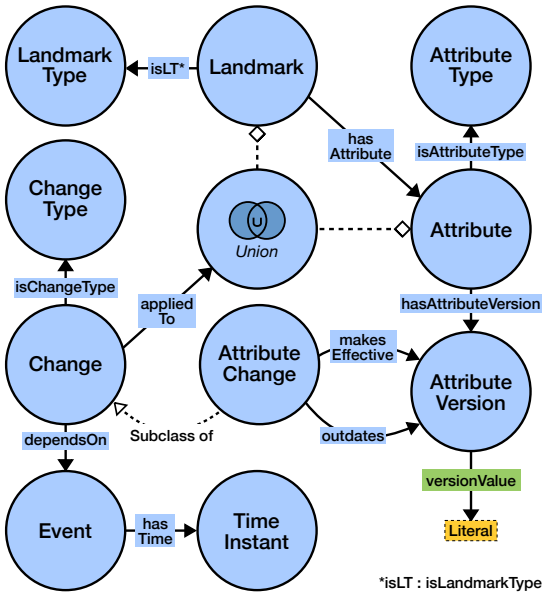

FIGURE 1 – Partie de l'ontologie PeGazUs pour modéliser l'évolution temporelle des entités géographiques.

Charles *et al.* [9] introduisent l'ontologie Hierarchical Historical Territory (HHT) et réutilisent aussi le concept de *Change Bridge* pour représenter l'évolution des unités territoriales de l'Ancien Régime et leurs multiples hiérarchies (religieuse, fiscale, judiciaire, etc.). Comme dans l'ontologie TSN-Change, cette évolution est représentée sous la forme de versions successives de ces unités territoriales, leurs propriétés demeurant constantes pour une même version. Dans le but de représenter de manière plus flexible les évolutions des emprises spatiales des unités territoriales, une extension de l'ontologie HHT a été ajoutée qui vise à la décrire sous la forme de zones élémentaires pouvant être intégrées à l'une ou l'autre des unités territoriales représentées selon les connaissances extraites des sources historiques [10].

L'ontologie PeGazUs [7] permet de décrire les entités géographiques et leurs évolutions (voir figure 1), mais se distingue des précédentes en versionnant non plus les entités géographiques, mais leurs attributs et en représentant les changements à leur niveau. Une entité géographique y est représentée dans la classe `Landmark`. On lui associe sa nature `LandmarkType` via le prédicat `isLandmarkType`. Elle possède des attributs (`Attribut`) typés (*cf.* `AttributeType`). Chaque attri-

1. Nomenclature des unités territoriales statistiques, voir https://ec.europa.eu/eurostat/fr/web/nuts/background

but compte un nombre non limité de versions définis par la classe `AttributeVersion`. Une version comporte une valeur qui est un `Literal`.
Un événement, associé à un instant (`TimeInstant`), décrit une évolution du territoire (fusion, scission, changement de nom, de géométrie, etc) entraînant une ou plusieurs modifications des ressources de type (`Landmark` ou `Attribut`) représentées par un changement. Par exemple, la disparition d'une entité géographique est décrite par un changement de type `LandmarkDisappearance`. `AttributeChange` est une sous-classe de `Change` et permet de décrire une évolution dans les données au niveau des attributs. Elle permet de préciser la façon dont le changement s'applique à l'attribut en décrivant la mise en effectivité (avec `makesEffective`) et/ou l'obsolescence (avec `outdates`) de versions.

2.2 Inférences des dynamiques spatio-temporelles à partir de descriptions d'états

Plusieurs approches ont été proposées dans la littérature [21, 6, 11], pour peupler des gazetiers historiques concernant des unités administratives ou statistiques. Les deux premières utilisent des jeux de données géographiques structurées versionnés par année de validité des données. La dernière permet d'intégrer des données associées à des dates de validité différentes au sein du même jeu de données, mais décrivant l'évolution des entités géographiques sans recouvrements ni lacunes. Seule l'approche proposée par [7] permet d'intégrer des données sur d'autres types d'entités géographiques, dont la généalogie est moins bien documentée dans les sources historiques et pour lesquelles on ne dispose généralement que d'attestations d'existence, discontinues dans le temps, présentant tantôt des recouvrements temporels tantôt des lacunes. Cependant, cette approche comporte encore des limites : elle ne permet pas de prendre en compte des données décrivant les événements qui affectent les entités géographiques ; les changements inférés peuvent présenter des incertitudes temporelles ; ils peuvent également présenter des incohérences en cas d'attestations contradictoires concomitantes.
Qu'elles utilisent les systèmes d'information géographique [15], ou les graphes de connaissances [21, 5], les premières approches proposées dans la littérature pour peupler un *gazetteer* historique utilisent des jeux de données géographiques structurées, versionnés par année de validité des données. Sur la base d'un critère d'identité donné, elles comparent les états successifs des entités géographiques deux à deux afin de détecter des changements potentiels entre eux et d'en déduire les types d'événements du monde réel qui ont pu conduire à de tels changement au niveau des données.
L'approche proposée par [11] s'en distingue en permettant de traiter des données géographiques associées à des dates de validité différentes au sein du même jeu de données, mais décrivant les états des entités géographiques concernées sans recouvrements ni lacunes temporels entre ces états. Cette approche a été récemment étendue pour détecter de qualifier des changements des emprises spatiales des unités territoriales représentées même lorsque les données de géométrie destinées à les représenter sont absentes ou imprécises. Utilisant le principe de blocs élémentaires pour constituer des géométries, cette approche est particulièrement adaptée dans le cas où les entités géographiques sont hiérarchisées par des relations d'inclusion spatiale comme c'est le cas pour les unités administratives. Cette approche se limite cependant à l'étude de l'évolution de géométries : elle ne peut être appliquée pour des attributs comme le nom, le code INSEE, etc.
La limite de toutes ces approches est qu'elles ne s'appliquent pas aux entités géographiques, dont la généalogie est moins bien documentée dans les sources historiques que dans le cas des unités administratives, et pour lesquelles on ne dispose que d'attestations d'existence, discontinues dans le temps, présentant tantôt des recouvrements temporels tantôt des lacunes. De plus, l'identité des entités géographiques y est soit connue a priori, soit triviale à retrouver. Ainsi, les faits qui décrivent la même entité sont bien associés à cette dernière dès leur intégration dans le graphe.
L'approche de peuplement d'un graphe de connaissances géo-historique à partir de données hétérogènes, fragmentaires et incomplètes provenant de sources multiples proposée par Bernard *et al.* [7] vise à dépasser ces limites. Chaque source historique contient des informations à intégrer qu'on appelle factoïdes [23]. Ces dernières sont représentées selon l'ontologie Pegazus et leurs triplets sont stockés dans un graphe nommé regroupant l'ensemble des factoïdes liés à la source. Le résultat final est un graphe de faits qui est la réconciliation de l'ensemble des graphes de factoïdes où chaque fait est sourcé par un ou plusieurs factoïdes. La première étape consiste à détecter les faits au sein des différentes sources qui décrivent les mêmes entités géographiques sur la base d'un ou plusieurs critères. Puis, les faits sont ordonnés temporellement afin d'en déduire les événements qui décrivent l'apparition et la disparition des entités géographiques qu'ils décrivent. Ce tri temporel ne repose pas sur l'algèbre des intervalles d'Allen [1] du fait de la présence d'instants dans la phase de tri et des possibles ambiguïtés dans la combinaison de relations, notamment en procédant à un raisonnement par transitivité. L'évolution des attributs de chaque entité géographique est ensuite inférée en détectant les changements entre deux versions d'attributs successives.
Si elle présente l'avantage de permettre d'intégrer des données fragmentaires extraites de sources historiques offrant des représentations partielles, discontinues et potentiellement redondantes ou complémentaires des entités géographiques au cours du temps, cette approche comprend plusieurs limites. Premièrement, les changements inférés sont temporellement flous. Si deux versions successives sont espacées d'un siècle, alors l'incertitude sur la date du changement sera de l'ordre du siècle. Au lieu d'utiliser uniquement des versions pour en déduire des changements comme le font toutes les méthodes, il serait donc utile d'intégrer des changements dans les données de départ qui permettraient d'inférer des versions. Enfin, la méthode a des problèmes

d'inférence de changements dans le cas où des versions différentes se chevauchent temporellement. Elle en déduit des événements dont la période temporelle associée est située dans un intervalle $[t_i, t_j]$ où $t_i > t_j$, ce qui est constitue une incohérence.

3 Reconstruction de l'évolution des entités

Dans cette section, nous introduisons une extension de PeGazUs [2] qui, en plus d'introduire une ontologie, propose une méthode de reconstitution de l'évolution des entités géographiques à partir de données hétérogènes et fragmentaires [7]. La méthode se décompose en trois grandes étapes. En premier lieu, il s'agit de représenter les données sur les entités géographiques conformément à l'ontologie PeGazUs puis de lier les données décrivant les mêmes entités en fixant un critère de similarité. Pour les quartiers et les voies de communication, ce critère repose sur la similarité du nom tandis pour les numéros d'habitation, deux entités sont équivalentes si elles possèdent le même numéro en plus d'être liés à la même voie par le biais d'une relation (`LandmarkRelation`) de même nature.
Puis, nous créons une succession de versions d'attributs élémentaires indivisibles qui ne se chevauchent pas temporellement et où il n'existe pas de période temporelle sans version d'attribut (voir figure 4). Enfin, nous fusionnons les versions successives initialisées lors de l'étape précédente selon plusieurs critères. Par exemple, deux versions successives de "Nom" dotées de labels identiques pour la même rue seront fusionnées.

3.1 Initialisation multi-source des entités

Comme le fait Bernard *et al.* [7] (voir section 2.2), il faut débuter par la description des données issues des sources selon notre ontologie. Les triplets générés sont dans des graphes nommés associés chacun à une source. Ils permettent la création du graphe de faits résultant de la construction multi-source du référentiel. Chaque fait est lié à un ou plusieurs factoïdes comme le montre la figure 2. Pour une ressource r du graphe des faits et une de ses attestations a dans une source donnée, il existe un triplet tel que $hasTrace(r, a)$. On dit que r est tracé par a.
Ce liage permet d'inférer les changements d'apparition et de disparition ainsi que les événements dont ils dépendent. Si, pour une entité géographique, il existe dans les sources des attestations de sa création (respectivement de sa disparition), alors on peut générer une instance de la classe `Change` de type `LandmarkAppearance` (respectivement `LandmarkDisappearance`) et déduire quand cette dernière est apparue (respectivement a disparu) à partir du temps valide associé à cette attestation dans la source. Dans le cas contraire, on ne peut pas avoir une valeur temporelle précise mais une estimation. En extrayant

2. *PErpetual GAZeteer of approach-address UtteranceS*. Sa documentation, les données et le script permettant de construire le graphe sont disponibles sur le dépôt https://github.com/charlybernard/pegazus-extension

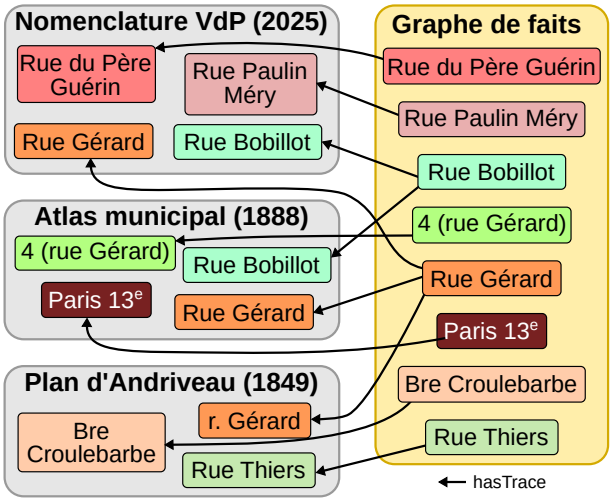

FIGURE 2 – Initialisation du graphe de faits à partir des graphes de factoïdes décrivant chacun une source.

le temps valide associé à la plus ancienne attestation d'une entité dans une source, on en déduit que son apparition se déroule avant la date extraite. On fait de même avec la disparition en sélectionnant la date de l'attestation la plus récente, dans la source considérée. En prenant l'exemple de la figure 2, la rue Bobillot est mentionnée dans une source datant de 1888 ainsi que dans une autre datant 2025. Ainsi, on peut déduire que la voie apparaît avant 1888 et disparaît après 2025.
Toutefois, l'apparition et la disparition ne sont pas les seuls changements qui s'appliquent aux entités géographiques. Les évolutions de noms ou de géométrie constituent d'autres changements à représenter. L'ontologie PeGazUs modélise les attributs de manière indépendante, ce qui permet de représenter leur évolution avec un ensemble de versions associées chacune à une période de validité délimitée par des changements. Néanmoins, ce processus demande une méthode spécifique dont la première partie demande d'effectuer une représentation élémentaire de leur évolution.

3.2 Représentation élémentaire de l'évolution des attributs

Après l'étape initiale, les attributs des entités géographiques agrègent différents factoïdes — des versions (`AttributeVersion`) et des changements (`AttributeChange`) — à partir desquels il est possible de reconstituer l'évolution de chaque attribut en générant des faits. Pour un attribut A d'une entité géographique, notons $C = \{c_1, ..., c_n\}$ l'ensemble des changements et $V = \{v_1, ..., v_m\}$ pour celui des versions.
Pour commencer, il faut procéder à un découpage élémentaire comme décrit par la figure 4. L'objectif est de générer une succession de versions élémentaires et indivisibles en fonction de l'ordre relatif des attestations. Pour cela, deux phases de traitement sont nécessaires : l'initialisation de

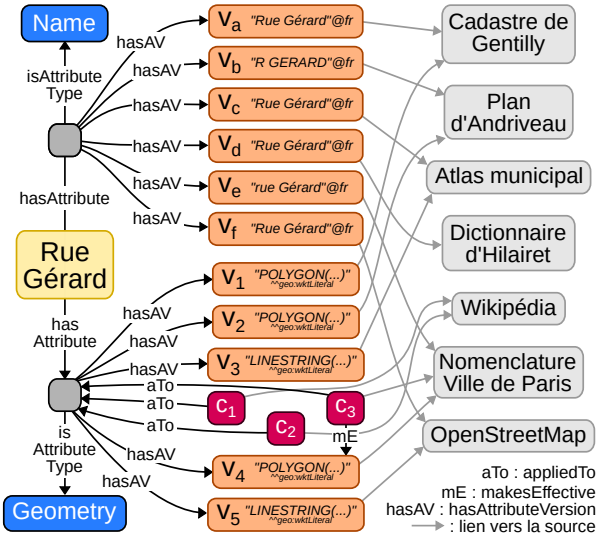

FIGURE 3 – Regroupement des attestations des différentes sources sur les attributs de la rue Gérard.

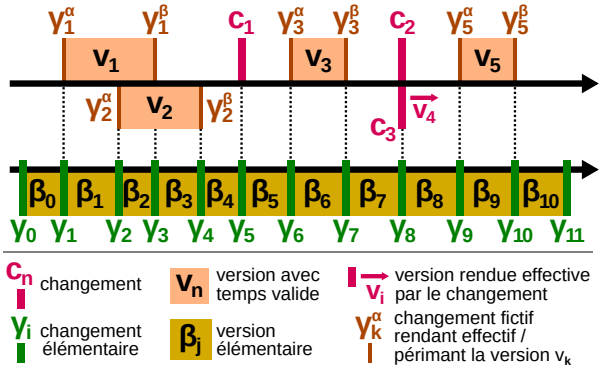

FIGURE 4 – Création de l'ensemble de changements Γ et de versions B (frise du bas) à partir de versions et de changements. La frise du haut montre l'agencement temporel des éléments de C et V pour l'attribut "géométrie" de la rue Gérard.

changements et de versions et l'affiliation de ces données initialisées aux factoïdes.

En prenant l'exemple de la rue Gérard située à Paris dont une représentation multi-source est donnée par la figure 3, son attribut "géométrie" est composé de huit factoïdes :

- une version v_1 valable sur la période 1845-1849, d'après le cadastre napoléonien de Gentilly ;
- une version v_2 valable entre 1847 et 1851, d'après le plan Andriveau de la ville de Paris ;
- une version v_3 valable entre 1887 et 1889, d'après l'atlas municipal de Paris ;
- deux changements de géométrie c_1 et c_2 qui ont respectivement lieu en 1857 et 1979 selon Wikipédia ;
- un changement de géométrie c_3 qui a lieu en 1979 et rend effectif la version v_4, d'après la nomenclature des voies de Paris ;
- une dernière version v_5 valable entre 2024 et 2025, d'après OpenStreetMap.

On remarque qu'il y a d'une part un chevauchement temporel des temps valides de v_1 et v_2 que, d'autre part, il existe des intervalles temporels conséquents pour lesquels nous ne disposons d'aucune information. Malgré ces lacunes et ces chevauchements, notre méthode permet tout de même de reconstruire l'évolution de la géométrie de la rue.

3.2.1 Initialisation des changements et des versions

Cette section consiste à associer à chaque attribut une succession de versions élémentaires indivisibles qui ne se chevauchent pas temporellement séparés par des changements comme le montre la frise du bas de la figure 4. Ces initialisations se font à partir des factoïdes associés à l'attribut (les éléments des ensembles C et V) en commençant par générer des changements élémentaires puis en inférant une version entre chaque paire de changements successifs.

Pour former un ensemble Γ de changements élémentaires, on définit $\Omega = C \cup C_V$ avec $C_V = \left\{\gamma_1^\alpha, \gamma_1^\beta, \ldots, \gamma_m^\alpha, \gamma_m^\beta\right\}$ où $\forall i \in [\![1; m]\!], makesEffective(\gamma_i^\alpha, v_i) \wedge outdates(\gamma_i^\beta, v_i)$. C_V est un ensemble de changements fictifs appliqués aux versions ayant un temps valide. Pour le cas de la rue Gérard, toutes les versions, exceptée v_4, ont un temps valide donc sont associés chacun à deux changements fictifs indiquant sa mise en effectivité et sa péremption. Pour v_1, il y a deux changements γ_1^α et γ_1^β associés respectivement aux valeurs temporelles 1845 et 1849. L'attribut "géométrie" de la rue est ainsi associé à $\Omega = \left\{c_1, c_2, c_3, \gamma_1^\alpha, \gamma_1^\beta, \gamma_2^\alpha, \gamma_2^\beta, \gamma_3^\alpha, \gamma_3^\beta, \gamma_5^\alpha, \gamma_5^\beta\right\}$. Les changements c_2 et c_3 ont lieu en 1979 donc leur simultanéité permet d'en déduire qu'ils sont similaires. En agrégeant les changements simultanés de Ω, on génère l'ensemble Γ valant $\{\gamma_1, \ldots, \gamma_{10}\}$.

On y ajoute deux changements $\gamma_{-\infty}$ et $\gamma_{+\infty}$ associés à deux instants infinis respectivement négatifs et positifs (que sont γ_0 et γ_{11} dans la figure 4). Enfin, on trie temporellement ces éléments en reliant deux changements successifs avec le triplet $hasNextChange(c_i, c_j)$. Pour $\gamma \in \Gamma$, si $\exists \delta = \underset{x \in \Omega \setminus \{\gamma\}, t(x)-t(\gamma)>0}{\arg\min} t(x) - t(\gamma)$ où $t(x)$ est la valeur temporelle associé au changement x alors $hasNextChange(\gamma, \delta)$.

Les changements maintenant initialisés et ordonnés temporellement, il est aisé d'en faire de même avec les versions. Une version est initialisée entre deux changements successifs générés lors de l'étape précédente. Ainsi, un ensemble de versions B est créé tel que : $\forall(\gamma_i, \gamma_j) \in \Gamma^2, hasNextChange(\gamma_i, \gamma_j) \implies \exists \beta \in B, AttributeVersion(\beta) \wedge makesEffective(\gamma_i, \beta) \wedge outdates(\gamma_j, \beta)$.

3.2.2 Affiliation des changements et des versions

Une fois les changements de Γ et les versions de B créés, il faut les affilier respectivement aux éléments existants de C et V : $\forall c \in C, \exists \gamma \in \Gamma, hasTrace(\gamma, c)$ (cela vaut aussi

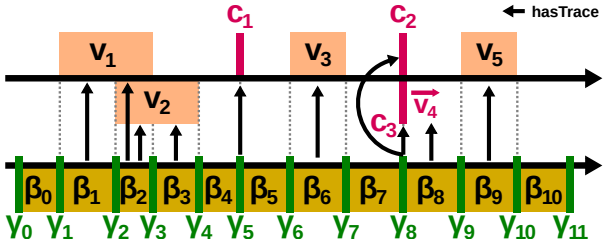

FIGURE 5 – Affiliation des changements et des versions pour l'attribut géométrie de la rue Gérard.

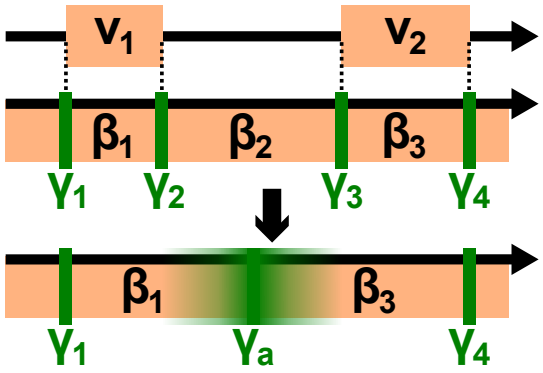

FIGURE 6 – Cas d'une version non tracée (ici β_2) délimitée par deux changements non tracés.

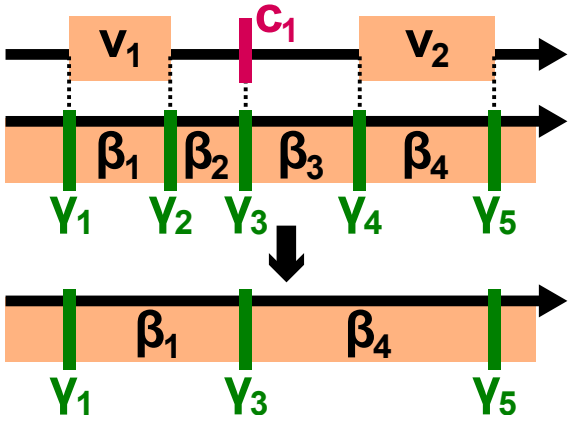

FIGURE 7 – Cas d'une version non tracée (ici β_2 et β_3) délimitée par un seul changement non tracé.

respectivement pour V et B). L'intérêt de l'affiliation est de lier les données que l'on vient d'initialiser aux factoïdes.
L'affiliation des changements de C à ceux de Γ se fait lors de l'étape présentée dans la section précédente : si un changement γ de Γ est créé à partir d'un changement c de C alors $hasTrace(\gamma, c)$. Ainsi, pour la rue Gérard, les triplets générés sont $hasTrace(\gamma_5, c_1)$, $hasTrace(\gamma_8, c_2)$ et $hasTrace(\gamma_8, c_3)$.
Pour les versions, l'affiliation est moins triviale car contrairement aux changements, il est possible qu'une version de V soit la trace de plusieurs versions de B. D'après la figure 5, β_1 et β_2 sont tracées par v_1 et β_2 et β_3 sont tracées par v_2. Pour $(v, \beta) \in (V, B)$, la relation $hasTrace(\beta, v)$ est satisfaite si l'une des deux conditions suivantes est remplie : soit si l'intersection de leurs temps valides est non vide, soit s'ils dépendent chacun d'un changement, et l'un de ces changements est la trace de l'autre. Autrement dit, $\exists (c, \gamma) \in (C, \Gamma)$ tel que $makesEffective(c, v) \wedge makesEffective(\gamma, \beta) \wedge hasTrace(\gamma, c)$. Cette condition reste valable si l'on remplace $makesEffective$ par $outdates$. Dans tous les cas, l'affiliation des versions se fait à partir de celles faites pour les changements. Concernant la rue Gérard, les triplets de type $hasTrace(\beta_i, v_j)$ sont générés pour les paires (β_1, v_1), (β_2, v_1), (β_2, v_2), (β_3, v_2), (β_6, v_3), (β_8, v_4) et (β_9, v_5).

3.3 Reconstitution de l'évolution des attributs à partir de leurs versions élémentaires

3.3.1 Suppression de versions lacunaires

Le découpage élémentaire présenté en section 3.2 permet d'obtenir le découpage le plus fin des changements et versions qui existent dans l'ensemble des sources pour un attribut d'une entité géographique. À cette étape, on est capable de reconstituer l'évolution d'une entité géographique selon ce que disent les sources. Toutefois, pour des attributs, il existe des périodes durant lesquelles aucune information venant des sources n'est fournie. Dans le cas de la rue Gérard, les versions β_0, β_4, β_5, β_7 et β_{10} ne sont pas tracées donc durant leur temps valide, on ne sait pas ce que vaut l'attribut "géométrie". Les supprimer en les fusionnant avec leur voisine (leur prédécesseure et/ou leur successeure) permettrait de combler les lacunes sur ces intervalles temporels. Différents critères doivent être pris en compte pour que les fusions se fassent sans perturber la cohérence des évolutions temporelles. Fusionner deux versions voisines implique de supprimer le changement qui les lie. Or, un changement tracé résulte d'une information sourcée : il n'est donc pas souhaitable de le supprimer. Ainsi, il sera impossible de fusionner deux versions voisines liées par un changement tracé. Par conséquent, les paires (β_4, β_5) et (β_7, β_8) de la figure 5 ne pourront être fusionnées.
Pour une version β non tracée, trois cas existent :

- ses deux changements ne sont pas tracés ;
- un seul des deux changements est tracé ;
- ses deux changements sont tracés.

Le premier cas est illustré par la figure 6. β_2 est une version dont les deux changements ne sont pas tracés donc on pourrait la fusionner avec β_1 ou β_3. Dans ce cas, il convient de supprimer β_2 et de fusionner ses deux changements. Autrement dit, il faut que le changement qui rend obsolète β_1 (et rend effectif β_2) soit le même que celui qui rend effectif β_3 (et périme β_2). Étant donné que ce cas implique la fusion de deux changements γ_1 et γ_2 ayant des valeurs temporelles t_1 et t_2 distinctes (avec $t_1 < t_2$, le changement γ résultant de l'agrégation n'a pas d'instant t précis, on peut juste en déduire que $t_1 \leq t \leq t_2$.

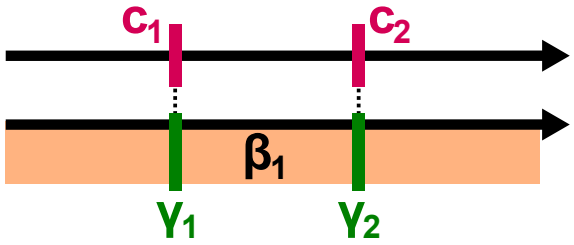

FIGURE 8 – Version non tracée (ici β_1) délimitée par deux changements tracés.

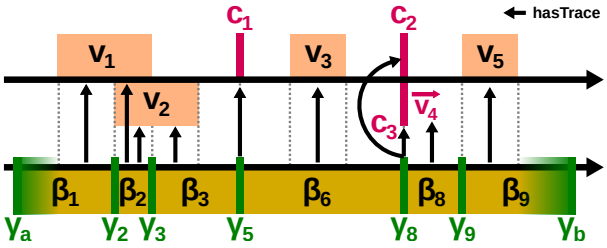

FIGURE 9 – Reconstitution de l'évolution temporelle après suppression de versions non tracées pour l'attribut géométrie de la rue Gérard.

Le deuxième cas, illustré par la figure 7, prend place lorsqu'un seul des deux changements liés à une version est tracé. Dans ce cas, il convient de le fusionner avec sa voisine, avec laquelle il partage un changement non tracé. Dans l'exemple de la figure 7, β_2 partage un changement non tracé avec β_1 tout comme β_3 avec β_4.

Lorsque les deux changements d'une version sont tracés (illustré par la figure 8), alors il ne faut rien faire.

Concernant l'attribut géométrique de la rue Gérard, les versions β_0 et β_{10} sont concernées par le premier cas. β_0 est supprimée et ses changements γ_0 et γ_1 sont fusionnés sous γ_a (voir figure 9). β_1 est ainsi rendue effective par γ_a dont la valeur temporelle est située entre $-\infty$ et 1845. Pour β_{10} qui est aussi supprimée, ses changements γ_{10} et γ_{11} fusionnent pour former γ_b. Ensuite, le deuxième cas s'applique aux versions β_4, β_5 et β_7. Tandis que β_4 est absorbée par β_3, les deux autres le sont par β_6. Enfin, aucune version n'est concernée par le dernier cas.

3.3.2 Fusion des versions élémentaires similaires successives

L'étape finale consiste à fusionner les versions similaires en fonction de leur valeur. L'objectif de la section précédente était de générer une alternance de versions et de changements sans chevauchements temporels. Cette génération dépend de l'agencement initial. Le résultat présenté sur la frise chronologique en figure 9 dépend de la manière dont sont triées les données fournies par la frise du haut. Le but de cette dernière étape est de prendre en compte les valeurs associées aux versions : deux versions successives dont les valeurs sont similaires sont potentiellement à fusionner.

Il est donc nécessaire, au préalable, de comparer les fac-

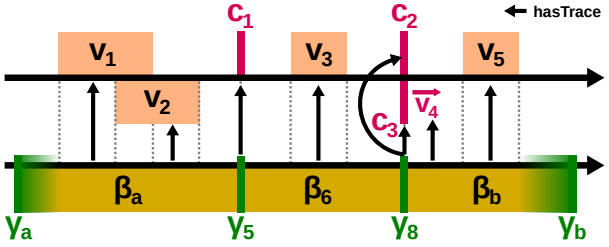

FIGURE 10 – Reconstitution finale de l'évolution temporelle après fusion des versions similaires pour l'attribut géométrie de la rue Gérard.

toïdes décrivant des versions entre eux en fonction de leur valeur via deux prédicats : `sameVersionValueAs` et `differentVersionValueFrom`. Les critères de similarité sont à personnaliser en fonction du type d'attribut considéré. Par exemple, nous avons choisi de comparer les versions d'attributs de nom par stricte égalité de leur nom simplifié. Ce nom simplifié est le nom de la voie auquel on fait les traitements suivants : suppression des signes diacritiques ; remplacement des caractères non alphanumériques par des espaces ; suppression des articles, prépositions, conjonctions et adverbes ; mise en bas de casse des caractères ; tri alphanumérique des caractères (avec la suppression des espaces).

Ainsi, pour une version dont la valeur est « rue du Père-Guérin », sa valeur simplifiée avant le tri alphanumérique sera « rue pere guerin » soit « eeeeginprrruu » après.

Du fait de l'hétérogénéité importante des géométries venant des sources, déterminer la similarité de deux versions v_1 et v_2 d'un attribut de type géométrie ayant des valeurs g_1 et g_2 est moins aisé. Nous proposons ici d'adopter la méthode suivante : en notant $S(g)$ la surface d'une géométrie, on peut considérer que v_1 et v_2 sont similaires si $\frac{S(g_1 \cap g_2)}{S(g_1 \cup g_2)} \geq \alpha_{min}$ où $\alpha_{min} \in [0;1]$ est un coefficient minimal de similarité.

Une fois ces comparaisons effectuées, il est possible de procéder aux fusions si nécessaire. Comme pour l'étape précédente, on ne peut pas fusionner deux versions successives si elles sont séparées par un changement tracé. Les conditions pour fusionner deux versions β_1 et β_2 sont les suivantes :

- avoir un changement en commun qui ne soit pas tracé : $\exists \gamma, \nexists hasTrace(\gamma, c) \wedge makesEffective(\gamma, \beta_1) \wedge outdates(\gamma, \beta_2)$;
- les versions dont elles dérivent sont similaires : $hasTrace(\beta_1, v_1) \wedge hasTrace(\beta_2, v_2) \wedge (sameVersionValueAs(v_1, v_2) \vee v_1 = v_2)$.

Si on reprend l'exemple de la rue Gérard, les comparaisons indiquent des similarités entre les versions v_1, v_2 et v_3 ainsi qu'entre v_4 et v_5. En appliquant les conditions de fusion, les versions tracées par v_1 et v_2 sont à fusionner ensemble, soit β_1, β_2 et β_3, via la version β_a (voir figure 10). Bien que similaire à v_2, la version v_3 n'est pas prise en compte ici car il existe le changement c_1 qui le sépare. Enfin, β_8 et β_9 sont à fusionner pour former β_b puisqu'elles sont respectivement

tracées par v_4 et v_5 et que $sameVersionValueAs(v_4, v_5)$.

Grâce à cette approche, nous sommes désormais capables de reconstituer l'évolution détaillée de chaque attribut d'une entité géographique, permettant de constituer l'historique de cette dernière. Cependant, pour garantir la fiabilité de ce processus, il est nécessaire d'évaluer la cohérence des données générées par la méthode présentée.

4 Vérification de la cohérence

Une façon d'évaluer la méthode de peuplement que nous proposons consiste à vérifier la cohérence du graphe des faits produit. Plusieurs éléments sont à vérifier :

- les factoïdes modélisés selon l'ontologie ;
- le graphe de faits construit selon l'ontologie ;
- l'alternance de versions et de changements ;
- l'absence de versions lacunaires, supprimées conformément lors de l'étape du traitement décrite en section 3.3.1 ;
- l'absence de versions successives similaires conformément à l'approche décrite dans section 3.3.2.

Une première étape pour s'assurer de la cohérence du graphe de faits consiste à s'assurer que les données à partir desquelles il est construit sont elles-mêmes cohérentes. Pour ce faire, nous testons la cohérence des triplets des graphes de factoïdes représentés avec l'ontologie PeGazUs à l'aide de règles SHACL [22] et de requêtes SPARQL. Ceci nous assure que la forme des données initiales ne peut pas être la raison d'incohérences dans le graphe final. Le même traitement est ensuite appliqué sur le graphe de faits pour vérifier que la méthode de peuplement que nous proposons produit bien un graphe compatible avec l'ontologie PeGazUs. Puis, nous vérifions l'alternance de versions et de changements pour les attributs dans le graphe de faits par l'intermédiaire d'une requête SPARQL. Elle permet aussi de vérifier qu'il n'existe pas de période temporelle non couverte par une version d'attribut conformément aux objectifs présentés en section 1. Pour s'assurer de l'absence de telles lacunes, on vérifie que toutes les versions sont tracées par au moins une affirmation venant des graphes de factoïdes. Si des versions non tracées existent, il convient de vérifier si les changements qui leur sont associés sont tracés comme décrit en figure 8. Dans ce cas, elles restent bien cohérentes. Enfin, l'évaluation de l'étape de fusion de versions successives similaires présentée en section 3.3.2 doit être réalisée manuellement. Pour simplifier cette vérification et éviter d'avoir à naviguer dans le graphe, un outil de visualisation cartographique. D'une part, il permet de visualiser l'évolution d'une entité géographique et celle de l'état du territoire à un instant donné. L'interface présente l'agencement des versions avec leur date de validité et leur(s) valeur(s) sur une frise chronologique. Pour les attributs de type géométrie, une carte interactive affiche les valeurs des versions. D'autre part, l'outil propose d'afficher l'état du territoire à un instant donné.

5 Mise en œuvre de la méthodologie

5.1 Application sur la Butte aux Cailles

Notre approche a été testée sur un ensemble de données provenant de sources décrivant les voies et numéros d'immeuble du quartier de la Butte aux Cailles situé dans le 13^{e} arrondissement de Paris sur une période allant de la fin du XVIII^{e} siècle jusqu'à aujourd'hui. Ce quartier était situé dans la commune de Gentilly avant 1860, date à laquelle le territoire parisien s'est étendu jusqu'à l'enceinte de Thiers. Cet événement a permis la transformation urbaine de ce quartier agricole en quartier résidentiel assez dense, ce qui implique de nombreux changements sur les entités géographiques de cette zone.

5.2 Présentation des sources utilisées

Pour reconstituer l'évolution des adresses et des rues de la Butte aux Cailles, nous avons utilisé des données contemporaines comme les *Dénominations des emprises des voies actuelles* [3] et les *Dénominations caduques des voies* [4] de la ville de Paris qui décrivent des voies de communication. La *Base Adresse Nationale* [5] (BAN) décrit des numéros d'immeubles. Des données d'OpenStreetMap, de Wikidata et de Wikipédia sont aussi prises en compte.
Nous avons également intégré des données vectorisées manuellement à partir de plans anciens décrivant le territoire parisien : le cadastre napoléonien de Gentilly (1847), le plan Andriveau de 1849 [3], le plan parcellaire municipal (1871) et l'atlas municipal de 1888 [2].

5.3 Évaluation du graphe final

Ne disposant pas de vérité terrain sur laquelle s'appuyer, nous ne sommes pas en mesure de produire une évaluation quantitative et systématique du graphe final. Nous pouvons toutefois faire une étude qualitative sur le graphe obtenu en le comparant avec l'historique des voies fournie par la nomenclature des *Dénominations des emprises des voies actuelles* de la ville de Paris, historique que nous n'utilisons pas en entrée du processus de peuplement du graphe. L'outil de visualisation mentionné en section 4 et illustré par la figure 11, fournit une frise chronologique pour chaque attribut de l'entité sélectionnée. En comparant les résultats de la reconstitution automatique de l'évolution des rues, on remarque que les données sont globalement en accord avec celles fournies dans cet historique. Néanmoins, il demeure quelques incohérences, causées par des conflits entre sources ou bien par des critères de similarité entre versions successives trop stricts (voir section 3.3.2). Par exemple, le graphe nous indique qu'un changement de nom aurait eu lieu le 9 août 1888 où la rue Bobillot devient la rue Bobillot : l'erreur ici est qu'avant cette date, il n'existait aucun nom pour la voie. Cette erreur s'explique par le fait que le plan parcellaire municipal de la ville de Paris a été établi

3. https://opendata.paris.fr/explore/dataset/denominations-emprises-voies-actuelles
4. https://opendata.paris.fr/explore/dataset/denominations-des-voies-caduques
5. https://adresse.data.gouv.fr/

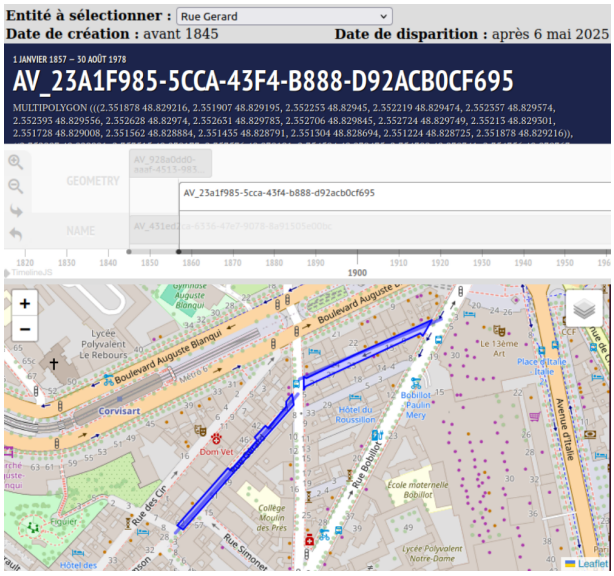

FIGURE 11 – Outil de visualisation de l'évolution des entités géographiques affichant la rue Gérard. La géométrie affichée est valable entre 1857 et 1978.

FIGURE 12 – Outil de visualisation de l'évolution des entités géographiques affichant un snapshot pour l'année 1888 autour de la rue Gérard. Le fond de plan est le plan d'Andriveau.

grâce à des relevés réalisés entre 1871 et 1896. Le temps valide des entités a été fixé ici autour de 1871 alors que le relevé pour cette rue est plus récent. Ici, c'est donc la source utilisée qui cause une incohérence dans les données, pas notre méthode de peuplement. À l'inverse, pour la rue Gérard, notre méthode déduit qu'il y a eu un changement de géométrie dans les années 1850 alors qu'aucune source ne le mentionne. Cette erreur est due à un problème de non fusion de versions similaires, estimées à tort comme non suffisamment similaires.

Parallèlement, l'outil de visualisation temporelle a été utilisé pour générer des snapshots du territoire à des dates spécifiques, sélectionnées en fonction de leur inclusion dans les intervalles de validité des sources utilisées pour la construction du graphe. Par exemple, l'année 1888 a été retenue afin d'évaluer la conformité des entités reconstruites avec le plan Andriveau de cette période.

6 Conclusion

Dans cet article, nous avons présenté une approche pour peupler un graphe de connaissances géo-historique d'adresses à partir de données hétérogènes et fragmentaires. La contribution de ce travail est la méthodologie de construction de l'évolution spatio-temporelle des entités géographiques avec des données décrivant des états ou des événements sans que les sources dont elles sont issues ne couvrent temporellement la totalité de la période d'existence des entités géographiques. À l'inverse, certaines sources utilisées peuvent présenter des chevauchements temporels et proposer des attestations contradictoires. Pour vérifier la cohérence des données en sortie, nous fournissons un ensemble de préconisations avant de mettre en œuvre la méthode sur un jeu de données décrivant le quartier de la Butte aux Cailles à Paris depuis la Révolution française.

Par la suite, nous comptons évaluer cette méthode de peuplement de façon quantitative et systématique, en l'appliquant sur des jeux de données d'adresses récents, exhaustifs, et à différents temps valides, dont on retire tour à tour l'un ou l'autre des millésimes, destiné à servir de vérité terrain pour son année de validité.

Les données représentant des événements utilisées dans cet article proviennent de données textuelles et leur extraction et structuration ont été réalisées manuellement. Une piste d'enrichissement serait de permettre leur reconnaissances et leur structuration automatiques à l'aide de grands modèles de langage (LLM). Les données et les scripts utilisés pour cet article sont disponible sur le dépôt https://github.com/charlybernard/pegazus-extension.

Références

[1] James F. Allen. Maintaining Knowledge about Temporal Intervals. In Daniel S. Weld and Johan de Kleer, editors, *Readings in Qualitative Reasoning About Physical Systems*, pages 361–372. Morgan Kaufmann, January 1990.

[2] Adolphe Alphand and Louis-François Sébastien Fauve. Atlas municipal des vingt arrondissements de la Ville de Paris dressé sous l'Administration de M. Ferdinand Duval, Préfet, sous la Direction de M. Alphand ; par les soins de M. L. Fauve, géomètre en chef, avec le concours des Géomètres du Plan de Paris, 1888.

[3] J Andriveau-Goujon. Plan de Paris fortifié et des communes environnantes : 1849 / Le plan et la lettre gravés par P. Rousset, 1849.

[4] Merrick Lex Berman, Ruth Mostern, and Humphrey Southall, editors. *Placing Names : Enriching and In-*

tegrating Gazetteers. Indiana University Press, August 2016.

[5] Camille Bernard, Christine Plumejeaud, Marlène Villanova-Oliver, Jerome Gensel, and Hy Dao. *An Ontology-based Algorithm for Managing the Evolution of Multi-Level Territorial Partitions*. November 2018.

[6] Camille Bernard, Marlène Villanova-Oliver, Jérôme Gensel, and Hy Dao. Ontologies pour représenter l'évolution des découpages territoriaux statistiques. *Revue Internationale de Géomatique*, 28(4) :409–437, December 2018.

[7] Charly Bernard, Solenn Tual, Nathalie Abadie, Bertrand Duménieu, Joseph Chazalon, and Julien Perret. PeGazUs : A Knowledge Graph Based Approach to Build Urban Perpetual Gazetteers. In Mehwish Alam, Marco Rospocher, Marieke Van Erp, Laura Hollink, and Genet Asefa Gesese, editors, *Knowledge Engineering and Knowledge Management*, volume 15370, pages 364–381. Springer Nature Switzerland, Cham, 2025. Series Title : Lecture Notes in Computer Science.

[8] Peter K Bol. The China Historical Geographic Information System (CHGIS) Choices Faced, Lessons Learned. *Conference on Historical Maps and GIS*, 23, August 2007.

[9] William Charles, Nathalie Aussenac-Gilles, and Nathalie Hernandez. HHT : An Approach for Representing Temporally-Evolving Historical Territories. In Catia Pesquita, Ernesto Jimenez-Ruiz, Jamie McCusker, Daniel Faria, Mauro Dragoni, Anastasia Dimou, Raphael Troncy, and Sven Hertling, editors, *The Semantic Web*, volume 13870, pages 419–435. Springer Nature Switzerland, Cham, 2023. Series Title : Lecture Notes in Computer Science.

[10] William Charles, Nathalie Aussenac-Gilles, and Nathalie Jane Hernandez. Diachronical geometry without polygons : the extended HHT ontology for heterogeneous geometrical representations. volume 15233, page 80. Springer Nature Switzerland ; Springer, November 2024.

[11] Jean Charlet, Bruno Bachimont, and Raphaël Troncy. Ontologies pour le Web sémantique. Technical report, CNRS, réseau thématique pluridisciplinaire Documents publié dans le numéro spécial Web sémantique de la revue I3 (https ://www.irit.fr/journal-i3/hors_serie/annee2004/index_fr.php), 2004.

[12] Christophe Claramunt, Marius Thériault, and Christine Parent. A qualitative representation of evolving spatial entities in two-dimensional topological spaces. In *Innovations In GIS 5*, pages 128–142. CRC Press, March 1998.

[13] Benoît Costes. *Vers la construction d'un référentiel géographique ancien : un modèle de graphe agrégé pour intégrer, qualifier et analyser des réseaux géo-historiques*. PhD Thesis, Université Paris-Est, November 2016.

[14] Géraldine Del Mondo. *Un modèle de graphe spatio-temporel pour représenter l'évolution d'entités géographiques*. phdthesis, Université de Bretagne occidentale, Brest, October 2011.

[15] Bertrand Dumenieu. *Un système d'information géographique pour le suivi d'objets historiques urbains à travers l'espace et le temps*. PhD Thesis, École des Hautes Études en Sciences Sociales, December 2015.

[16] Y. T. Fan, J. Y. Yang, D. H. Zhu, and K. L. Wei. A time-based integration method of spatio-temporal data at spatial database level. *Mathematical and Computer Modelling*, 51(11) :1286–1292, June 2010.

[17] Ian N. Gregory, Chris Bennett, Vicki L. Gilham, and Humphrey R. Southall. The Great Britain Historical GIS Project : From Maps to Changing Human Geography. *The Cartographic Journal*, 39(1) :37–49, June 2002.

[18] Karl Grossner, Krzysztof Janowicz, and Carsten Keßler. Place, Period, and Setting for Linked Data Gazetteers. In Merrick Lex Berman, Ruth Mostern, and Humphrey Southall, editors, *Placing Names*, The Spatial Humanities, pages 80–96. Indiana University Press, Bloomington, IN, 2016.

[19] Pierre Hallot. *L'identité à travers l'espace et le temps. Vers une définition de l'identité et des relations spatio-temporelles entre objets géographiques.* PhD Thesis, ULiège - Université de Liège, March 2012.

[20] Kathleen Hornsby and Max J. Egenhofer. Identity-based change : a foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, 14(3) :207–224, April 2000.

[21] Tomi Kauppinen, Jari Väätäinen, and Eero Hyvönen. Creating and Using Geospatial Ontology Time Series in a Semantic Cultural Heritage Portal. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *The Semantic Web : Research and Applications*, volume 5021, pages 110–123. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. Series Title : Lecture Notes in Computer Science.

[22] Holger Knublauch and Dimitris Kontokostas. Shapes Constraint Language (SHACL), July 2017.

[23] Michele Pasin and John Bradley. Factoid-based prosopography and computer ontologies : towards an integrated approach. *Digital Scholarship in the Humanities*, 30(1) :86–97, April 2015.

Extraction et interprétation sémantique de tables anciennes : défis et perspectives

Solenn Tual[1], Nathalie Abadie[1], Joseph Chazalon[2], Bertrand Duménieu[3], Julien Perret[1]

[1] LASTIG, Université Gustave Eiffel, IGN-ENSG
[2] LRE, EPITA
[3] CRH, Ecole des Hautes Etudes en Sciences Sociales

{solenn.tual, nathalie-f.abadie, julien.perret}@ign.fr, joseph.chazalon@epita.fr, bertrand.dumenieu@ehess.fr

Résumé

Les documents historiques contenant des tables représentent une source d'informations précieuse dans divers domaines. Si les institutions patrimoniales numérisent massivement ces documents pour en faciliter l'accès, les connaissances structurées qu'ils contiennent demeurent difficilement accessibles faute de pouvoir être requêtées. Cet article propose une revue des méthodes d'extraction d'informations dans des tables historiques numérisées et d'interprétation sémantique de tables tout en identifiant leurs limites. Les défis et perspectives associés à chaque tâche sont identifiés afin de proposer une chaîne de traitement visant à extraire et à structurer les informations contenues dans des tables historiques sous la forme de graphes de connaissances.

Mots-clés

Interprétation sémantique de tables, Graphes de connaissances, Extraction d'informations, HTR, NER, Documents historiques.

Abstract

Historical documents containing tables represent a valuable source of information in various fields. Although heritage institutions are digitising these documents on a massive scale to facilitate access, the structured knowledge they contain remains difficult to access because it cannot be queried. This article reviews methods for extracting information from digitised tables and for semantically interpreting tables, while identifying their limitations. The challenges and perspectives associated with each task are identified in order to propose a processing chain aimed at extracting and structuring the information contained in historical tables in the form of knowledge graphs.

Keywords

Semantic table interpretation, Knowledge graphs, Information extraction, HTR, NER, Historical documents.

1 Introduction

Les fonds d'archives regorgent de documents contenant des tables, formes privilégiées d'organisation et de présentation des données prisées par les administrations, grandes productrices de registres. On en retrouve également communément dans les ouvrages imprimés, notamment à visée scientifique, comme dans des documents manuscrits divers tels que livres de comptes, contrats, index géographiques.

Longtemps peu valorisés par les institutions patrimoniales, cette masse de documents historiques tabulaires constitués d'une feuille à des dizaines de milliers de pages est aujourd'hui graduellement diffusée sous forme numérisée. Cette diffusion se limite toutefois généralement à un partage sur le Web des images des pages numérisées, sans possibilité d'interroger les tables qu'elles contiennent.

La sémantique d'une table est essentiellement portée par l'organisation tabulaire elle-même : les lignes guident le regroupement de l'information en unités cohérentes, tandis que les colonnes apportent une information sémantique fine à l'échelle des champs. Les différents éléments d'un tableau et l'information qu'ils contient sont caractérisées par une grande densité de mentions d'entités et des relations riches. Reconnaître et préserver cette organisation particulière tout en extrayant sa sémantique constitue donc un enjeu d'accès et de valorisation pour ces fonds d'archives. C'en est aussi un, majeur, pour les sciences sociales quantitatives, les grands corpus de documents historiques tabulaires restant largement sous-exploités en raison des coups prohibitifs de traitements qu'ils impliquent.

Faciliter l'accès aux connaissances structurées en tables et contenues dans des documents historiques tabulaires est l'enjeu de cet article.

Il existe des travaux d'extraction et représentation sémantique des informations contenues dans des documents historiques sous la forme de graphes de connaissances [27, 21, 36, 14]. Cependant, les tables anciennes ne font pas partie des types de documents considérés, ces derniers étant principalement des textes en prose, confrontés à des problématiques d'extraction d'informations qui leur sont propres.

L'extraction en masse des informations contenues dans des

documents historiques tabulaires est confrontée à de nombreux défis : forte variabilité de la structure des tables, mélange de textes manuscrits et imprimés, présence d'abréviations ou encore usage de vocabulaires spécifiques. En outre, une fois le texte brut de ces tables transcrit, il ne donne qu'une vue limitée des connaissances disponibles dans l'intégralité de la collection. Annoter sémantiquement les tables d'un corpus offre la possibilité de lier les informations à travers les pages et les documents pour constituer des bases de connaissances sérielles et cohérentes ouvrant la perspective d'analyses longitudinales à grande échelle.

L'association des méthodes d'extraction d'informations (IE) dans des documents et d'interprétation sémantique de tables (STI) semble particulièrement pertinente pour parvenir à cet objectif. Aussi, après un passage en revue des caractéristiques des tables dans les documents historiques (section 2), nous présentons les tâches et des méthodes d'extraction d'informations dans de tels documents (section 3). Nous décrivons ensuite les principales tâches et approches d'interprétation sémantique de tables (section 4). Après avoir identifié les perspectives et les verrous scientifiques restants pour exploiter conjointement ces deux disciplines, nous proposons une chaîne de traitement de documents historiques tabulaires réconciliant les techniques d'annotation à des fins d'extraction d'informations dans des images et celles d'interprétation sémantique de tables à l'aide d'un graphe de connaissances de domaine (section 5).

2 Tables dans les documents historiques

Les données considérées dans cet article sont des tables issues de documents historiques. Une table est une structure bidimensionnelle composée de n lignes et m colonnes. La cellule est le plus petit élément à l'intersection d'une ligne et d'une colonne. Un document historique est numérisé à partir d'une source physique papier, imprimée ou manuscrite, conservée dans un service d'archives, un musée ou encore une bibliothèque. Ces documents sont généralement décrits par des métadonnées qui fournissent des éléments de contexte souvent indispensables à leur compréhension. Les types de tables présentes dans des documents, historiques ou non, sont très variés. Les tables peuvent être classées dans deux catégories [24] : *les tables de mise en forme*, utilisées pour structurer du contenu sans cohérence sémantique, et les *tables classiques*, caractérisées par une forte cohérence entre les lignes et les colonnes et qui contiennent des connaissances interprétables. Nous ne nous intéressons ici qu'aux tables classiques pour lesquelles il est possible d'utiliser des systèmes d'interprétation sémantique de tables. Ces tables classiques peuvent être décrites selon trois dimensions [24], qui, quand elles sont combinées, permettent de définir finement différents types de tableaux :

- la **structure** : tables imbriquées ou divisées, cellules fusionnées, cellule contenant des énumérations de valeurs ;
- les **relations internes** entre les cellules, les lignes et les colonnes : tables relationnelles, tables d'entités, matrices ;
- l'**orientation** : tables horizontales, tables verticales, matrices.

Pour décrire des tables issues de documents historiques, cette taxonomie doit cependant être complétée par des éléments propres aux sources anciennes. Les tables considérées dans cet article sont originellement des documents "papier" — manuscrits, imprimés ou préimprimés et complétés manuellement — qui ont été conservés puis numérisés. Elles n'existent pas nativement dans un format numérique, comme c'est le cas des tables présentes sur le Web et traditionnellement considérées pour l'interprétation sémantique de tables. Ainsi, la **nature du document historique**, **sa structure et la place de la table dans celui-ci** ainsi que son **état de conservation** sont autant de paramètres qui modifient la compréhension de la table et l'accès aux informations qu'elle contient. Les types de documents historiques qui contiennent des tables sont très variés (cf. Figure 1). Par exemple, les registres utilisés pour le recensement ou le cadastre, les journaux de bord, les ouvrages scientifiques et les manuels contiennent des tables. La source peut être composée d'un seul document ou faire partie d'une collection plus vaste. Les registres issus d'une même collection ayant existé sur une longue période, comme le cadastre, contiennent des informations similaires dans des tables dont la structure varie d'une administration à l'autre et a évolué au cours du temps (cf. Figure 1 (a) et (b)). La table peut constituer l'élément principal du document ou servir de complément à d'autres contenus, tels que du texte ou des illustrations. Elle peut s'étendre sur plusieurs pages, constituer le corps principal du document ou alors ne couvrir qu'une partie d'une page. Les colonnes de la table sont généralement réparties sur une page simple ou sur une double-page. Un registre peut contenir une série de tables élémentaires (présentées sur une page simple ou double), susceptibles d'être agrégées en une seule en raison de leur structure commune. La fragmentation d'une table en plusieurs pages (et donc images) risque néanmoins de compliquer la reconstitution de sa structure. Par exemple, la reliure du registre est susceptible de masquer les colonnes voisines (et donc le texte). Elle peut également entraîner un décalage entre les pages d'une double-page, désalignant ainsi les cellules d'une même ligne. La structure initiale d'une table est souvent préimprimée. La complétion manuscrite des documents permettait aux scripteurs d'ajouter des lignes, colonnes ou cellules supplémentaires qui complexifient la lecture du document. L'état de conservation du document physique impacte l'accès à l'intégralité ou non des informations contenues dans la table. Par exemple, un registre qui contient une table distribuée dans plusieurs pages ne sera pas complet si une page est manquante ou que son contenu est illisible. Lors de la numérisation, il est essentiel de conserver l'ordre des pages afin de préserver la logique de lecture des informations.

Ainsi, bien que fortement structurées, les tables issues de documents historiques restent difficiles à exploiter dans leur totalité. Le texte qu'elles renferment ne devient acces-

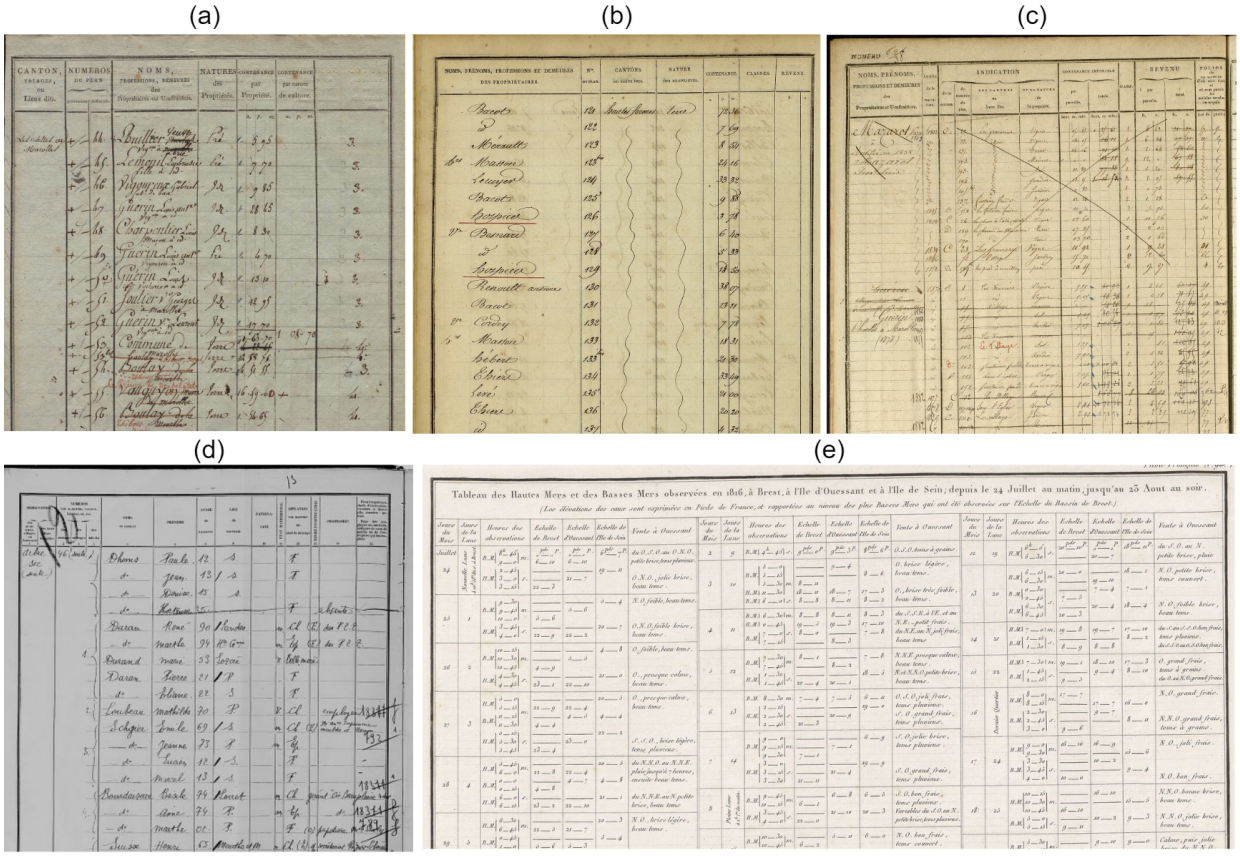

FIGURE 1 – **(a)** Page de registre d'états de sections de Marolles-en-Brie (1810) : table relationnelle horizontale. L'identité du sujet (parcelle) est partiellement masquée : l'identifiant de la section dans laquelle se trouve la parcelle est absent. Il faut se référer à la page de couverture du chapitre pour connaître l'identifiant de la section. **(b)** Page de registre d'états de sections d'Ivry (date inconnue, ultérieure à 1822) : même type de table que (a) mais l'ordre des colonnes est différent. **(c)** Matrice des contribuables de Marolles-en-Brie (1822-1914) : matrice, la colonne relative aux contribuables (première colonne ici) contient des cellules avec énumérations. Une page peut contenir plusieurs sous-tableaux relatifs aux propriétés de chaque contribuable. **(d)** Page de registre du recensement de Paris (1926). Des groupes de lignes doivent être formé pour reconstituer les ménages. **(e)** Tableau des hautes et basses mers observées en 1816 dans la région de Brest, extrait du Pilote français (tome 1, 1822)
Sources : Archives Départementales du Val-de-Marne, (a) 3P 387, (b) 3P 1631 et (c) 3P 389, (d) Archives de Paris D2M8 221, (e) Bibliothèque nationale de France, département Cartes et plans, GE CC-1194

sible aux machines qu'après une phase de transcription. Il doit être structuré pour permettre une recherche par attribut. Une fois cette étape franchie, les données obtenues ne fournissent que des informations partielles, souvent complexes à interroger et à croiser au sein d'une même collection. L'annotation sémantique du texte brut extrait constitue donc une approche pertinente pour valoriser pleinement le contenu de ces documents historiques.

3 Extraction d'informations dans des documents tabulaires historiques

L'extraction d'informations dans des documents numérisés a pour objectif de localiser, transcrire et organiser le texte qu'ils contiennent dans le but de traduire un contenu initialement exclusivement sous forme graphique vers une représentation informatique structurée exploitable automatiquement. Selon les approches, les étapes qui constituent ce processus peuvent être distinctes ou au contraire intégrées, comme c'est le cas dans les approches les plus modernes. En effet, ce domaine connaît des avancées significatives grâce au progrès de l'apprentissage profond.

3.1 Tâches et chaîne de traitement

Nous décrivons ci-dessous les principales tâches d'un processus d'extraction d'informations dans des documents historiques numérisés.

La classification des images permet d'identifier, au sein d'un répertoire de documents numérisés, celles qui nécessitent un traitement approfondi. En effet, les informations à extraire se trouvent souvent sur un type spécifique de pages. Lorsqu'elles sont disponibles, les métadonnées associées aux images décrivent généralement l'ensemble du répertoire sans distinction. Il est donc nécessaire de classer

les pages en fonction de leur nature (par exemple : couverture, tableau principal, récapitulatif intermédiaire, résumé) afin de ne conserver que celles pertinentes pour les étapes suivantes de la chaîne de traitement.

La reconnaissance de la mise en page du document consiste à comprendre la structure du document dans l'image. Par exemple, si l'image en entrée est une double-page où chaque page contient une partie des lignes d'une table, il faut détecter chaque page simple puis les différents éléments qui la composent (ligne ou colonne de la table, segment de texte).

La reconnaissance du texte (OCR/HTR) consiste à transcrire le texte. Pour des documents imprimés, les modèles utilisés sont des modèles d'OCR (*Optical Character Recognition*) alors que pour du texte majoritairement manuscrit ce sont des modèles d'HTR (*Handwritten Text Recognition*) qui sont utilisés. Le texte produit par ces systèmes contient généralement des erreurs de transcription, désignées sous le terme de "bruit".

La structuration des informations produites est le formatage final des données en sortie de la chaîne de traitement. Elle peut comprendre une phase de post-traitement du texte extrait et plus rarement d'enrichissement sémantique (reconnaissance des entités nommées, géocodage). Elle comprend également la production d'une base de données ou d'un fichier de données structurées (CSV, XML).

3.2 Approches

Il existe deux grands types de systèmes d'extraction d'informations : les approches classiques composées d'un enchaînement de modèles indépendants et les approches dites *end-to-end* où un même modèle réalise plusieurs étapes en une fois. Pour ces deux types d'approches, les régions contenant les tables sont généralement isolées au préalable grâce à un système de classification des pages ou de détection des régions.

Approches classiques Les approches classiques d'extraction d'informations dans des tables anciennes numérisées sont fondées sur un enchaînement de modèles où les résultats produits à une étape sont utilisés en entrée de l'étape suivante.

Après une phase de prétraitement des images (segmentation des double-pages en pages, redressage), *Constum et al. (2022)* [7] détectent la table dans la page, puis utilisent un modèle de segmentation pour extraire les lignes (sous la forme d'images) et enfin de les transcrire à l'aide d'un modèle d'HTR [10]. Un caractère spécial est utilisé pour matérialiser implicitement la séparation de la ligne en cellules. La structuration du contenu extrait est réalisée en post-traitement. La nature de chaque cellule est déduite de sa position dans la ligne. Ceci est possible car l'ordre des colonnes est connu *a priori* des auteurs et car la structure de l'ensemble des registres considérés ne varie pas.

Petit-Pierre et al. (2023) [28] proposent, à l'inverse, de détecter les colonnes et les segments de texte dans la page puis de regrouper ces segments en lignes *a posteriori*. Le texte est transcrit avec le modèle d'HTR proposé par *Puigcerver (2017)* [29].

Granell et al. (2023) [18] traitent des relevés météorologiques consignés dans des registres de navigation. La tâche de segmentation des lignes est réalisé par un modèle de classification des pixels [30]. La transcription du texte est réalisée à l'échelle de la ligne avec un modèle de type *Convolutionnal Recurrent Neural Networks* (CRNN).

Pour traiter des tables astronomiques datant du XIV^{e} au XVI^{e} siècle, *Eberle et al. (2024)* [15] détectent les chiffres dans l'image et les transcrivent avec un modèle d'HTR entraîné pour reconnaître uniquement les chiffres allant de 0 à 9, puis de recomposer les nombres correspondants à la valeur de chaque cellule en post-traitement.

Le principal défaut des approches avec segmentation préalable des documents est que les erreurs de découpage des pages en lignes ou colonnes se propagent aux étapes suivantes. Par ailleurs, une fois la segmentation effectuée, les modèles qui réalisent les tâches suivantes ne peuvent s'appuyer que sur un contexte visuel et textuel réduit [3]. Enfin, des jeux de données d'évaluation doivent être produits pour chaque étape.

Approches end-to-end *Boilet et al. (2024)* [3] traitent les registres du recensement de différents départements français. La chaîne de traitement comprend une étape de classification des pages numérisées avec YOLOv8, suivie de la transcription structurée du contenu des pages entières de tables avec le modèle Document Attention Network (DAN) [11], un encodeur convolutif suivi d'un décodeur Transformer [37]. Cette approche permet de retrouver les informations de même type sans segmenter l'image ni matérialiser la position exacte des lignes et des colonnes. Les annotations sont réalisées selon un modèle "Clé-Valeur" [35] : pour chaque zone de l'image (ligne), la liste des informations à extraire (correspondant à des colonnes ou à des entités nommées) est définie.

Constum et al. [9] introduisent DANIEL, une architecture dérivée de DAN [11] destinée à capturer l'organisation hiérarchique de l'information. Il a été testé pour extraire le contenu d'actes d'états civils manuscrits mais présente des perspectives intéressantes pour le traitement de tables.

Parallèlement aux approches spécialisées sur un type de document particulier, certaines approches telles que DONUT [22] ont visé à proposer des modèles entraînés sur un grand nombre de tâches afin de pouvoir s'adapter rapidement à de nouveaux problèmes ou types de documents. Ces approches ont été améliorées avec la réutilisation de grands modèles de langage (LLMs) existants dans la construction de modèles vision-langage (VLMs), disposant alors des capacités de raisonnement supérieures, mais montrant jusqu'à présent une performance limitée dans l'analyse de tables modernes, comme noté par *Scius-Bertrand et al.* [33] mi-2024. Tout récemment, le modèle GOT-OCR [39] a réalisé de grands progrès dans l'interprétation d'images de documents à la densité de texte élevée, ainsi que dans l'analyse de tables modernes, permettant leur transcription et plus simplement la réponse à des questions simples, laissant présager de nouvelles avancées dans l'analyse de do-

cuments historiques à court terme. Cependant, aucune approche n'arrive encore à s'affranchir d'un entraînement spécifique sur les documents historiques à analyser.

3.3 Synthèse et limites

Dans le domaine de l'extraction d'informations à partir de documents historiques, les modèles Transformer [37] ont favorisé l'émergence d'approches dites *end-to-end*. Celles-ci offrent de nouvelles perspectives : l'extraction d'informations peut être réalisée pour des pages entières sans segmentation préalable ni perte d'information contextuelle. Le texte transcrit est enrichi par des caractères spéciaux qui traduisent la typographie ou même la structure du document. Les jeux de données d'entraînement ne nécessitent pas de localisation de chaque élément du tableau dans l'image. Contrairement aux approches classiques, il n'est plus indispensable de constituer un jeu de données spécifique pour chaque tâche (détection de zones, transcription) pour entraîner les modèles. Un seul jeu de données complet, dont les annotations sont des fichiers de textes structurés traduisant le contenu et la structure de l'image, est suffisant. Le mécanisme d'attention [37] permet de localiser les informations dans l'image *a posteriori*. Ce type d'approches génère du contenu structuré lisible par une machine, traduction numérique du tableau représenté dans l'image. Cependant, ces approches ciblent principalement le traitement des tables relationnelles classiques. Les tables à la structure plus complexe, telles que les matrices, demeurent peu étudiées dans la littérature, de même que l'extraction de texte multi-orienté, le fait d'avoir un sens de lecture uniforme restant une dimension importante lors de la transcription. Par ailleurs, les architectures Transformers sont parfois sujettes à des hallucinations. Elles entraînent la génération de résultats erronés (mais souvent crédibles), tels que l'ajout de lignes supplémentaires dans une table. Les chaînes de traitement d'IE appliquées aux tables de documents historiques vont rarement au-delà de la structuration des données sous la forme de bases de données relationnelles [28, 7, 18, 3]. Ceci ne permet pas d'établir les liens entre des mentions d'entités identiques. Par ailleurs, les requêtes se font directement sur les chaînes de caractères qui sont souvent impactées par des erreurs du modèles de transcription utilisé.

4 Interprétation sémantique de tables

L'interprétation sémantique de tables (STI) est un processus qui consiste à annoter les différents éléments qui composent un tableau (ligne, colonne, cellule, tableau entier) et leurs relations à l'aide des ressources d'un graphe de connaissances (KG) pour améliorer l'interprétation de leur sémantique. L'objectif est de faciliter le développement d'applications mettant en œuvre les données qu'elles contiennent, comme la recherche d'informations, la réponse aux questions ou bien la création de bases de connaissances. Ces dernières années, les approches visant à résoudre les différentes tâches de STI se sont multipliées, stimulées par des compétitions comme SemTab [1]. Ces compétitions ont mené à une définition précise des tâches et des méthodes d'évaluation. Les deux principales applications de la STI sont d'enrichir les données représentées dans une table à l'aide de connaissances supplémentaires et de créer ou de compléter des KGs avec des informations issues de ces tables. Les approches proposées dans la littérature traitent généralement des fichiers tabulaires (CSV) ou des tables HTML.
L'étude des tâches, des étapes et approches d'interprétation sémantique de tables présentées ici est une synthèse des articles d'état de l'art de *Liu et al. (2023)* [24] et *Cremashi et al. (2024)* [13]. *Liu et al. (2023)* établissent une taxonomie détaillée des types de tables et de métadonnées, une présentation des tâches de STI et des graphes de connaissances généralistes. Les jeux de données d'évaluation, les métriques et une synthèse des grandes étapes de la chaîne de traitement sont passés en revue ainsi que les différents types d'approches développées pour réaliser les tâches recensées (jusqu'en 2021). Une comparaison des performances des principales approches pour effectuer chaque type de tâche est réalisée. *Cremashi et al. (2024)* [13] proposent une nouvelle taxonomie très détaillée des tâches et approches de STI (de 2007 à octobre 2024) reposant sur 31 critères ainsi qu'une comparaison des outils existants. Les travaux les plus récents utilisant les modèles de langages pré-entraînés (comme BERT) et les grands modèles de langages (LLMs) sont pris en compte. La chaîne de traitement est finement détaillée, incluant davantage d'étapes que dans l'article de *Liu et al. (2023)* [24].
Nous renvoyons à ces publications le lecteur qui souhaite entrer dans le détail des approches de STI.

4.1 Tâches

L'annotation de colonnes avec des types (CTA) consiste à attribuer un type à chaque colonne de la table. La distinction entre les colonnes correspondant à des entités, qui seront associées à des classes de l'ontologie, et les colonnes correspondant à des valeurs alphanumériques (dates, nombres), est particulièrement utile pour la réalisation de cette tâche [13].

L'annotation de colonnes avec des propriétés (CPA) vise à associer une propriété du KG à une paire de colonnes pour caractériser leur relation.

L'annotation de cellules avec des entités (CEA) consiste à associer le contenu d'une cellule (aussi appelée mention) à une entité du KG.

La détection des cellules qui décrivent de nouvelles entités (CNEA) est une tâche définie plus récemment par *Cremaschi et al. (2024)* [13]. Elle consiste à détecter des mentions comme étant des entités absentes du graphe de connaissances [26] utilisé pour l'annotation. Dans la littérature, ces cellules sont souvent annotées avec le terme *NIL (Not In Lexicon)*.

La thématisation consiste à associer un concept du KG à l'ensemble de la table ;

1. https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

La correspondance ligne-instance consiste à annoter une ligne de la table avec une entité du KG considérée comme le sujet de cette ligne. De ce fait, cette tâche est surtout pertinente dans le cas où la table considérée est une table relationnelle.
La tâche de CEA est particulièrement utile pour désambiguïser des entités mentionnées dans une table. Elle peut être assimilée à la tâche d'*Entity Linking* dans le domaine du traitement automatique du langage naturel [13] (TALN). Les tâches de CTA et CPA permettent de transformer automatiquement ou semi-automatiquement des tables en reliant leur structure au schéma de données défini par une ontologie [13].

4.2 Chaîne de traitement

Cette section présente les étapes d'un processus complet d'interprétation sémantique de tables tel que décrit par *Cremashi et al. (2024)* [13].

La préparation des données comprend une phase de standardisation des formats de tables, de nettoyage et de normalisation du contenu des cellules et de compréhension du type de table considéré. Les types de traitements appliqués dépendent des valeurs alphanumériques contenues dans la table. Par exemple, il peut s'agir de correction d'erreurs de frappes [5], de suppression de contenus inutiles [1], d'uniformisation de la casse, de conversion des unités de mesures [16, 12], d'expansion des abréviations [1] ou de conversion de formats de dates ou de coordonnées géographiques [31]. Ces prétraitements facilitent et améliorent la qualité de l'interprétation des tables considérées ;

La classification des colonnes vise à distinguer les colonnes dont les cellules contiennent des entités du graphe et celles qui correspondent à des valeurs alphanumériques (chaîne de caractères, nombre, date) [41, 20]. Cette étape permet de sélectionner les colonnes pour lesquelles les tâches de CEA et de CTA doivent être réalisées.

L'annotation des colonnes contenant des valeurs alphanumériques a pour objectif de déterminer le type des données contenues dans les colonnes qui ne contiennent pas d'entités (dates, nombres et leurs unités, coordonnées géographiques) [5, 12]. Cette étape constitue une partie de la tâche de CTA.

La détection du sujet vise à déterminer la colonne de la table contenant l'entité qui est aussi le sujet de la ligne [4, 17]. Cette étape est particulièrement utile pour la tâche de CPA, car elle identifie la colonne racine de nombreuses propriétés [13].

La résolution d'entités correspond aux tâches de CEA et CNEA. La CEA peut être décomposée en trois sous-tâches : la détection des mentions dans la table, l'identification des entités du KG candidates au liage et des mentions sans entités correspondantes dans le KG ainsi que la désambiguïsation d'entités. La détection de mentions nécessite souvent une structuration plus fine du contenu d'une cellule, par exemple avec une étape de reconnaissance des entités nommées (NER) [13]. L'identification des entités candidates revient à sélectionner un sous-ensemble de ressources du KG et à les indexer suivant divers critères. Enfin, la désambiguïsation vise à départager des entités candidates très similaires. Le contexte s'avère particulièrement utile pour résoudre cette sous-tâche.

La prédiction des types des colonnes contenant des entités est le résultat final de la CTA. Cette étape peut comprendre une phase de présélection des types des entités de chaque colonne ou être le résultat de prédictions d'un algorithme entraîné avec des données similaires [31].

La prédiction des prédicats associés à des paires de colonnes est le résultat final de la CPA.

4.3 Approches

Il existe trois grandes familles d'approches d'interprétation automatique de tables [24]. Elles peuvent être utilisées pour traiter les différentes étapes et tâches de la chaîne de traitement décrites précédemment.

Les approches heuristiques Pour associer un élément de la table à la ressource du KG jugée la plus pertinente, les approches heuristiques s'appuient sur les métriques et les critères de décisions usuels en recherche d'information : similarités, TF-IDF, vote majoritaire ou encore méthodes probabilistes. Ces approches sont utilisées pour résoudre les tâches de CEA, CTA et de CPA. Elles peuvent introduire ou non une phase de reclassement des candidats qui améliore généralement les résultats.

L'ingénierie de caractéristiques repose sur l'extraction de caractéristiques lexicales et statistiques des éléments du tableau qui sont ensuite utilisées pour entraîner des modèles d'apprentissage classique (SVM, K-NN ou Random Forest). La tâche la plus traitée par ces méthodes est la CTA. Ces méthodes nécessitent de disposer de jeux de données pour entraîner les modèles utilisés.

L'apprentissage profond est basé sur l'entraînement de réseaux de neurones profonds. Nous pouvons distinguer deux groupes de méthodes [24] : l'apprentissage de plongements représentant les éléments de la table (cellules, lignes, colonnes) afin de les comparer dans l'espace vectoriel et l'apprentissage de plongement des entités du graphe. Comme pour l'ingénierie des caractéristiques, des jeux de données doivent être produit pour entraîner les modèles. Les méthodes basées sur l'utilisation des LLM connaissent actuellement de multiples développements [40].

4.4 Synthèse et limites

Malgré les avancées récentes, il demeure encore de nombreux défis à résoudre pour annoter sémantiquement des tables complexes comme celles issues de documents historiques. Comme pour l'IE, la plupart des approches se concentrent sur l'annotation de tables à la structure simple, principalement des tables d'entités ou tables relationnelles horizontales à sujet monocellulaire. Or, il existe une grande diversité de tables aux structures plus complexes, notamment parmi les tables de documents historiques. Nous identifions par exemple des structures imbriquées ou à sujets cachés ou composés (identifiant à reconstituer à partir de plusieurs colonnes ou à l'aide du contexte). La grande ma-

jorité des approches de STI existantes ont été évaluées avec des jeux de donnés d'état de l'art, comme T2Dv2 [32] et Limaye [23]. Ceci est utile pour comparer les approches entre elles, mais ne considère pas l'adaptation à des tables originales. Par ailleurs, les méthodes existantes utilisent principalement des graphes de connaissances encyclopédiques comme Wikidata [38], DBpedia [25], ou YAGO [34] pour l'annotation. Ces KG ne sont pas pertinents pour traiter des documents historiques qui contiennent des entités inconnues de ces bases. Les approches qui considèrent l'incomplétude des graphes de connaissances qu'elles utilisent sont encore rares [13]. Cette hypothèse est pourtant indispensable pour annoter des registres anciens qui décrivent des entités majoritairement non encyclopédiques. Enfin, l'utilisation du contexte de la table, de ses métadonnées est également insuffisamment développée alors qu'elle est identifiée comme bénéfique aux différentes tâches [24].

5 Interprétation sémantique de tables historiques : défis et perspectives

Nous pouvons définir l'interprétation sémantique de tables historiques comme un processus qui combine l'extraction d'informations dans des documents tabulaires historiques et l'annotation sémantique de tables dans une même chaîne de traitement. La revue des travaux existants montre qu'il n'existe pas de chaîne de traitement intégrant des approches de ces deux domaines pour extraire et gérer des connaissances contenues dans les tables anciennes. Leur articulation est pourtant particulièrement pertinente pour exploiter et interroger ce type de sources historiques en détail. Des connaissances isolées dans des documents tabulaires historiques peuvent être mises en relation dans un graphe afin de parcourir une collection de documents non plus d'images en images, mais d'entités en entités. Ceci facilite notamment la recherche d'informations dans des collections d'images de très grandes tailles.

Dans cette section, nous proposons une chaîne de traitement combinant extraction d'informations et interprétation sémantique de tables anciennes dans une perspective de peuplement de graphe de connaissances historiques puis nous recensons les défis qui demeurent pour y parvenir. Dans cette chaîne de traitement dont les étapes sont décrites ci-après, on suppose qu'une ontologie est déjà disponible pour représenter les connaissances contenues dans les tables.

La collecte des tables consiste à réunir les documents numérisés issus d'une ou plusieurs collections, accompagnés de leurs métadonnées. Ces métadonnées fournissent un contexte global sur les tables à traiter et peuvent être restructurées [3] afin d'en faciliter l'exploitation, notamment lorsque leurs structures varient. Ce sont des sources de connaissances privilégiées pour thématiser les tables à traiter et ainsi faciliter leur classification à l'étape suivante.

La classification des images est une étape récurrente dans les chaînes d'IE étudiées. Elle est nécessaire pour identifier les images qui contiennent les tables à traiter. Elle peut bénéficier d'une étape de thématisation préalable des répertoires d'images à l'aide de leurs métadonnées. La classification des images (pages de couvertures, résumés intermédiaires, synthèses) est utile pour fournir davantage de contexte aux tables et parfois pour compléter les informations qu'elles contiennent. Ces connaissances additionnelles peuvent venir compléter la thématisation initiale réalisée à l'aide des métadonnées.

La reconnaissance de la structure du tableau et du texte, réalisée avec des approches *end-to-end* utilisant des architectures Transformer comme DAN [11] ou DANIEL [8] permet d'extraire le contenu de chaque page sans segmentation des images et sans localisation du texte au préalable. Grâce à ce fonctionnement intégré, ces approches évitent la propagation d'erreurs d'une tâche d'IE à l'autre tout en permettant un gain de temps significatif pour produire des jeux de données d'entraînement [3] ; la production de données intermédiaires n'étant plus nécessaire.

En outre, ces approches permettent de résoudre, manuellement, les tâches de CTA et CPA lors de la définition du modèle d'annotation des images. En effet, l'utilisateur souhaite généralement extraire un ou plusieurs attributs des objets décrits dans la table. Ces attributs correspondent très souvent aux colonnes et sont généralement décrits sous la forme de classes dans une ontologie de domaine produite à partir de connaissances expertes des documents et du domaine auquel ils appartiennent. Aussi, les clés du modèle d'annotation "Clé-Valeur" utilisé pour produire les jeux de données d'entraînement de DAN correspondent à des classes de l'ontologie ou à des valeurs alphanumériques attendues, tandis que les valeurs du modèle d'annotation "Clé-Valeur" correspondent quant à elles, pour une ligne donnée, au contenu des cellules situées dans les colonnes considérées.

Par ailleurs, le mécanisme d'attention d'architectures telles que DAN permet d'absorber les variations d'emplacement et d'intitulés des colonnes qui contiennent les mêmes informations dans des tables appartenant à une même collection.

Le cas des colonnes qui peuvent contenir plusieurs types d'entités ou de valeurs alphanumériques constitue, en revanche, un défi qui reste à traiter. L'identification automatique du type des colonnes et des propriétés associées, dans le cas de documents aussi spécifiques que les documents historiques, reste également non résolue ce qui limite les possibilités de traitement massif de document tabulaires anciens sans intervention humaine.

Notons cependant que la qualité de la transcription fournie par le modèle peut avoir des conséquences importantes sur les étapes d'interprétation sémantique réalisées en aval de la chaîne de traitements.

La structuration des informations produites est très probablement l'étape qui peut bénéficier le plus des apports des approches d'interprétation sémantiques de tables. Celles-ci permettent en effet d'enrichir la transcription brute produite à l'étape précédente à l'aide d'approches de détection de sujet, de CEA et de CNEA permettant de proposer une structuration fine des données en sortie qui s'abstrait de celle de

la table en entrée, en particulier de l'ordre des colonnes, et facilite leur exploitation future.

Le cas de tables relationnelles, où le sujet correspond à une ligne, est certainement le plus simple à traiter. Il est cependant possible que l'identification unique du sujet ne soit réalisable qu'en intégrant des connaissances de thématisation produites lors des étapes de collecte et de classification des images ou en combinant les valeurs de plusieurs colonnes.

Les tâches de CEA et de CNEA permettent à cette étape de lier le texte brut produit par le modèle d'IE à des ressources du KG (notamment des concepts de taxonomies) ou de créer des entités qui vont être ajoutées au KG. Ces entités pourront ultérieurement être utilisées pour annoter de nouvelles tables. La principale difficulté pour leur mise en œuvre réside dans l'automatisation du choix de la méthode la plus appropriée selon le type d'entités ou de valeurs alphanumériques présents dans les différentes colonnes de la table transcrite.

De la même façon, dans le cas de nouvelles entités à intégrer au KG et à structurer au préalable, une étape de reconnaissance des entités nommées peut s'avérer nécessaire pour structurer des descriptions de personnes par exemple. L'automatisation du paramétrage de cette étape reste un verrou pour son exécution automatique sur de nombreux documents, sans intervention humaine.

L'identification automatique des entités mentionnées à plusieurs reprises dans une même table peut bénéficier des approches de détection de coréférence dans des documents [2] qui semblent particulièrement pertinentes pour rassembler les mentions d'un même objet qui apparaîtraient à plusieurs reprises et créer la ressource RDF correspondante selon le schéma de données décrit dans l'ontologie. Une piste intéressante pour réduire les erreurs liées à l'homonymie serait de contraindre la détection de coréférence à un sous-ensemble cohérent de documents au sein d'une même collection, par exemple dans un même registre ou un même espace géographique.

Lien aux sources La réalisation automatique des tâches d'IE et de STI permet ainsi de traiter de grandes collections d'images et de structurer les informations qu'elles contiennent sous la forme d'un graphe de connaissances. Cependant, maintenir le lien entre cette représentation dérivée du contenu de l'image et l'image elle-même demeure indispensable. L'image transmet des éléments de contexte utiles aux chercheurs en sciences humaines et sociales. Elle permet par ailleurs de confronter les informations lues aux données produites lors du processus automatique. Il est donc indispensable de développer des outils qui facilitent l'exploration des données par tous types d'utilisateurs. Le standard IIIF (*International Image Interoperability Framework*) [6] offre un ensemble de techniques pour diffuser des annotations d'images structurées selon le *Web Annotation Data Model* [19]. Ce standard est très utilisé par les services d'archives, les bibliothèques et les musées. Aussi, il serait particulièrement pertinent de visualiser les fragments de pages sur les images et d'afficher leur contenu enrichi sémantiquement (texte brut et ressources RDF associées, identifiées par URIs dans le graphe) dans une même interface. Le format de fichier employé pour la diffusion des annotations IIIF est le JSON-LD. Faisant partie des standards des Linked Open Data, il s'avère particulièrement adapté pour établir un lien entre la localisation des zones dans les images et les annotations structurées sous forme de ressources RDF générées par cette chaîne de traitement.

6 Conclusion

Cet article propose un passage en revue des tâches et principales approches d'extraction d'informations dans des tables historiques et d'interprétation sémantique de tables. Ces deux disciplines alliées ensemble offrent de nombreuses perspectives pour extraire, structurer et rendre accessible les connaissances contenues dans les tables de documents historiques numérisés en masse par les services d'archives ces dernières années. Après avoir identifié les limites des méthodes existantes, nous décrivons les étapes d'une chaîne de traitement spécialisée pour extraire et lier les données extraites des documents tabulaires historiques. Ces données sont structurées suivant une ontologie produite par des experts du domaine et des documents considérés. Le peuplement de l'ontologie avec les informations extraites des images facilite leur interrogation et leur mise en relation. Nous proposons également de développer des outils de visualisation et de requêtes du graphe de connaissances qui maintienne le lien aux sources numérisées et permette de déceler des erreurs. Le standard IIIF nous semble particulièrement pertinent pour mettre en œuvre cet objectif.
Cette approche ouvre de nouvelles perspectives pour la valorisation et l'exploitation de documents tabulaires historiques. Les travaux futurs viseront à évaluer cette chaîne de traitement sur un cas concret : les registres du cadastre napoléonien.

Remerciements

Ce travail a été soutenu financièrement par le Ministère des Armées – Agence de l'innovation de défense.

Références

[1] N. Abdelmageed and S. Schindler. JenTab : A Toolkit for Semantic Table Annotations. In *Proc. 2nd Int. Workshop on Knowledge Graph Construction*, volume 2873 of *CEUR Workshop Proceedings*, 2021.

[2] A. Arora, E. Silcock, M. Dell, and L. Heldring. Contrastive Entity Coreference and Disambiguation for Historical Texts. In *Proc. 2024 Conf. on Empirical Methods in Natural Language Processing*, pages 6174–6186, 2024.

[3] M. Boillet, S. Tarride, Y. Schneider, B. Abadie, L. Kesztenbaum, and C. Kermorvant. The Socface Project : Large-Scale Collection, Processing, and Analysis of a Century of French Censuses. In *ICDAR*, pages 57–73, 2024.

[4] Y. Chabot, T. Labbe, J. Liu, and R. Troncy. DAGOBAH : An End-to-End Context-Free Tabular Data Se-

mantic Annotation System. In *SemTab@ISWC*, pages 41–48, 2019.

[5] S. Chen, A. Karaoglu, C. Negreanu, T. Ma, J.-G. Yao, J. Williams, A. Gordon, and C.-Y. Lin. LinkingPark : An Integrated Approach for Semantic Table Interpretation. In *SemTab@ ISWC*, pages 65–74, 2020.

[6] IIIF Consortium. IIIF : How It Works. https://iiif.io/get-started/how-iiif-works/. Accessed : 2025-02-27.

[7] T. Constum, N. Kempf, T. Paquet, P. Tranouez, C. Chatelain, S. Brée, and F. Merveille. Recognition and Information Extraction in Historical Handwritten Tables : Toward Understanding Early 20th Century Paris Census. In *Document Analysis Systems*, pages 143–157, 2022.

[8] T. Constum, P. Tranouez, and T. Paquet. DANIEL : a fast document attention network for information extraction and labelling of handwritten documents. *IJDAR*, pages 1–23, 2025.

[9] Thomas Constum. *Extraction d'information dans des documents historiques à l'aide de grands modèles multimodaux*. Phd thesis, Normandie Université, France, 2024.

[10] D. Coquenet, C. Chatelain, and T. Paquet. End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1) :508–524, 2022.

[11] D. Coquenet, C. Chatelain, and T. Paquet. DAN : A Segmentation-Free Document Attention Network for Handwritten Document Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(7) :8227–8243, 2023.

[12] M. Cremaschi, A. Rula, A. Siano, and F. De Paoli. Semantic Table Interpretation Using MantisTable. In *Proc. 14th Int. Workshop on Ontology Matching*, volume 2536 of *CEUR Workshop Proceedings*, pages 195–196, 2019.

[13] M. Cremaschi, B. Spahiu, M. Palmonari, and E. Jimenez-Ruiz. Survey on Semantic Interpretation of Tabular Data : Challenges and Directions, 2024. arXiv :2411.11891.

[14] C. Díaz, J. Dunstan, L. Etcheverry, A. Fonck, A. Grez, D. Mery, J. L. Reutter, and H. R. Corral. Automatic Knowledge-Graph Creation from Historical Documents : The Chilean Dictatorship as a Case Study. In *Joint Proc. 2nd Workshop on KBC from PTLM (KBC-LM 2024) and 3rd Challenge on LM for KBC (LM-KBC 2024)*, volume 3853 of *CEUR Workshop Proceedings*, 2024.

[15] O. Eberle, J. Büttner, H. El-Hajj, G. Montavon, K.-R. Müller, and M. Valleriani. Historical insights at scale : A corpus-wide machine learning analysis of early modern astronomic tables. *Science Advances*, 10(43), 2024.

[16] B. Ell, S. Hakimov, F. Kaupmann, P. Braukmann, L. Cazzoli, A. Mancino, J. A. Memon, K. Rother, A. Saini, and P. Cimiano. Towards a Large Corpus of Richly Annotated Web Tables for Knowledge Base Population. https://pub.uni-bielefeld.de/record/2912802, 2017. dataset.

[17] S. Gottschalk and E. Demidova. Tab2KG : Semantic table interpretation with lightweight semantic profiles. *Semantic Web*, 13(3) :571–597, 2022.

[18] E. Granell, V. Romero, J. R. Prieto, J. Andrés, L. Quirós, J. A. Sánchez, and E. Vidal. Processing a large collection of historical tabular images. *Pattern Recognit. Lett.*, 170, 2023.

[19] W3C Web Annotation Working Group. Web Annotation Data Model. https://www.w3.org/TR/annotation-model/, February 2017. W3.org.

[20] T. Guo, D. Shen, T. Nie, and Y. Kou. Web Table Column Type Detection Using Deep Learning and Probability Graph Model. In *Web Inf. Syst. and Appl.*, pages 401–414, 2020.

[21] N. Jain, A. Sierra-Múnera, M. Lomaeva, J. Streit, S. Thormeyer, P. Schmidt, and R. Krestel. Generating Domain-Specific Knowledge Graphs : Challenges with Open Information Extraction. In *Proc. 1st Int. Workshop on Knowledge Graph Generation From (Text2KG 2022)*, volume 3184 of *CEUR Workshop Proceedings*, pages 52–69, 2022.

[22] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, and S. Park. Ocr-free document understanding transformer. In *ECCV*, pages 498–517.

[23] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2), 2010.

[24] J. Liu, Y. Chabot, R. Troncy, V.-P. Huynh, T. Labbé, and P. Monnin. From tabular data to knowledge graphs : A survey of semantic table interpretation tasks and methods. *J. Web Semant.*, 76 :100761, 2023.

[25] P. Mendes, M. Jakob, and C. Bizer. DBpedia : A Multilingual Cross-domain Knowledge Base. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1813–1817, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[26] C. Möller. Knowledge Graph Population with out-of-KG Entities. In *The Semantic Web : ESWC 2022 Satellite Events.*, volume 13384, pages 199–214, 2022.

[27] V. Nundloll, R. Smail, C. Stevens, and G. Blair. Automating the extraction of information from a historical text and building a linked data model for the domain of ecology and conservation science. *Heliyon*, 8(10) :e10710, 2022.

[28] R. Petitpierre, M. Kramer, and L. Rappo. An end-to-end pipeline for historical censuses processing. *IJDAR*, 26(4) :419–432, 2023.

[29] J. Puigcerver. Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition ? In *ICDAR*, pages 67–72, 2017.

[30] L. Quirós. P2pala : Page to page layout analysis tookit. https://github.com/lquirosd/P2PaLA, 2017.

[31] S. K. Ramnandan, A. Mittal, C. A. Knoblock, and P. Szekely. Assigning Semantic Labels to Data Sources. In *ESWC*, pages 403–417, 2015.

[32] D. Ritze and C. Bizer. Matching Web Tables To DBpedia - A Feature Utility Study. In *Proc. 20th Int. Conf. on Extending Database Technology (EDBT)*, 2017.

[33] A. Scius-Bertrand, A. Fakhari, L. Vögtlin, D. Ribeiro Cabral, and A. Fischer. Are layout analysis and OCR still useful for document information extraction using foundation models ? In *ICDAR*, pages 175–191, 2024.

[34] F. M. Suchanek, M. Alam, T. Bonald, L. Chen, P-H. Paris, and J. Soria. YAGO 4.5 : A Large and Clean Knowledge Base with a Rich Taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 131–140, Washington DC USA, July 2024. ACM.

[35] S. Tarride, M. Boillet, and C. Kermorvant. Key-Value Information Extraction from Full Handwritten Pages. In *ICDAR*, pages 185–204, 2023.

[36] S. Tual, N. Abadie, B. Duménieu, J. Chazalon, and E. Carlinet. Création d'un graphe de connaissances géohistorique à partir d'annuaires du commerce parisien du 19 ème siècle : application aux métiers de la photographie. In *IC*, 2023.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *arXiv :1706.03762*, 2017.

[38] D. Vrandečić and M. Krötzsch. Wikidata : a free collaborative knowledgebase. *Commun. ACM*, 57(10) :78–85, September 2014.

[39] H. Wei, C. Liu, J. Chen, J. Wang, L. Kong, Y. Xu, Z. Ge, L. Zhao, J. Sun, Y. Peng, C. Han, and X. Zhang. General OCR theory : Towards OCR-2.0 via a unified end-to-end model.

[40] T. Zhang, X. Yue, Y. Li, and H. Sun. TableLlama : Towards Open Large Generalist Models for Tables. In *NAACL*, pages 6024–6044, 2024.

[41] Z. Zhang. Effective and efficient Semantic Table Interpretation using TableMiner+. *Semantic Web*, 8(6) :921–957, 2017.