Représentation sémantique de questions pour interroger le Web sémantique.

Romain Beaumont, Brigitte Grau, Anne-Laure Ligozat

Représentation sémantique de questions pour interroger le Web sémantique

Romain Beaumont[1,2], Brigitte Grau[1,3] et Anne-Laure Ligozat[1,3]

[1] LIMSI-CNRS, France [2] Université Paris-Sud, France [3] ENSIIE, France
{romain.beaumont, brigitte.grau, anne-laure.ligozat}@limsi.fr

RÉSUMÉ. *Les bases de connaissances du Web sémantique sont généralement représentées sous forme de triplets RDF formant un graphe. Leur interrogation passe par un langage de type SPARQL, langage non maîtrisé des utilisateurs non experts, qui requiert de connaître le schéma de la base. C'est pourquoi les systèmes d'interrogation en langage naturel se développent actuellement. Se pose alors le problème de construction automatique de requêtes, devant intégrer des problèmes de distance lexicale entre les mots de la question et les relations de la base de connaissances. Dans cet article, nous proposons une nouvelle méthode d'analyse des questions qui opère par transformation de graphes, fondée sur des contraintes très générales sur la structure des requêtes, et qui résout les ambiguïtés sémantiques au plus tard par interrogation de la base. Nous proposons également une méthode d'identification des relations en nous fondant sur l'utilisation de WordNet. Nous obtenons de bons résultats pour l'identification de relations et des résultats prometteurs pour le système global, évalué sur une tâche de QALD3.*

ABSTRACT. *The knowledge base of the semantic Web are often represented by RDF triple repository that form a graph. It requires to use a dedicated language as SPARQL for interrogating them, that is generally not known by non-expert users. Moreover, it also require to know the knowledge base schema. To overcome these problems, the development of question answering systems in natural language is currently growing. In this paper, we propose a new method for the semantic analysis of questions, based on graph transformations, using very general constraints on the request structure, and that keeps ambiguities that are solved globally at the final step by interrogating the knowledge base. We also developed a method for dealing with lexical gap for identifying databse relations based on WordNet. We obtain good results on relation identification, and the global system show promising results.*

MOTS-CLÉS : *Web sémantique$_1$, système de question-réponse$_2$, analyse sémantique de question$_3$.*

KEYWORDS: *Semantic Web$_1$, Question-Answering system$_2$, Semantic question analysis$_3$.*

1. Introduction

De plus en plus de bases de connaissances sont disponibles sur le Web, telles que DBPedia (Auer *et al.*, 2007), Freebase (Bollacker *et al.*, 2008) ou Yago2 (Hoffart *et al.*, 2011), et se pose la question de pouvoir donner à un utilisateur un accès simple à ces connaissances. Ces bases sont généralement constituées de triplets (sujet, prédicat, objet), décrites en RDF (Resource Description Framework) qui peuvent être représentées comme des graphes, où sujets et objets sont les noeuds et les prédicats sont les arcs étiquetés par des labels. On parlera aussi de relation entre deux entités. Ces graphes peuvent être interrogés par un langage de requête, tel que SPARQL, qui reste difficile à maîtriser pour un utilisateur non expert. De ce fait, donner à un utilisateur la possibilité de profiter des outils du Web sémantique, tout en masquant leur complexité, amène à la conception d'interfaces faciles à utiliser. La conception de systèmes de question-réponse en langage naturel sur des données structurées s'est récemment développée dans ce but.

La génération de requêtes en SPARQL requiert 1) d'identifier les entités et relations de la base mentionnés dans la question, 2) de les associer en triplets et 3) de construire la requête elle-même, en déterminant sa structure et les opérateurs, tels que fonctions de calcul ou d'ordonnancement par exemple à lui adjoindre. La première tâche pose le problème des variations lexicales entre les labels associés aux entités et relations de la base et les termes employés par l'utilisateur, puisque celui-ci n'est pas guidé par la connaissance du schéma de la base. Se pose aussi le problème de résolution d'ambiguïtés sémantiques, car un même terme peut faire référence à différents objets ou prédicats. Par exemple, le verbe *married to* peut faire référence aux relations *dbo:spouse, dbo:partner, dbp:wife, dbp:husband, dbp:union, dbp:relationship*. Dans l'exemple suivant, *Which daughters of British earls died in the same place they were born in ?*, on peut trouver la réponse grâce aux entités de type *yago :DaughtersOfBritishEarls*, alors qu'on pourrait aussi considérer *daughter* comme une relation et *British earls* comme une entité. La deuxième tâche vise à construire une représentation sémantique de la question, et savoir rattacher les arguments aux prédicats auxquels ils doivent être liés. La construction de la requête elle-même est une traduction de la représentation sémantique.

Différentes solutions ont été apportées à ces problèmes. (Fader *et al.*, 2013) ne traitent que des questions simples qui correspondent à un seul triplet et realisent leur appariement par des techniques d'apprentissage. Afin de traiter des questions complexes, (Unger *et al.*, 2012) ont défini des patrons de requête qui guident l'analyse des questions, et résolvent ensuite l'identification des entités alors que (Yahya *et al.*, 2013 ; He *et al.*, 2014) partent de toutes les interprétations possibles, et construisent une représentation qui tient compte de contraintes de différentes natures, par une méthode fondée sur la programmation linéaire en nombre entier (ILP) ou sur les réseaux logiques de Markov (MLN). Ces méthodes ont peu considéré le problème de variation linguistique énoncé plus haut, et utilisent pour la plupart PATTY (Nakashole *et al.*, 2012) afin de reconnaître les relations. Cette ressource a été construite à partir d'articles Wikipedia par généralisation de graphes de dépen-

dances syntaxiques pour obtenir des patrons sémantiques de relation. Elle contient des variations des relations de DBpedia, mais seulement pour un nombre réduit de ces relations.

Nous proposons une méthode permettant de mieux identifier toutes les relations d'une base de connaissance en s'appuyant sur WordNet.

En ce qui concerne la résolution des ambiguïtés, nous avons choisi de ne pas les résoudre au moment de l'identification des éléments de la base mais de les conserver afin de pouvoir appliquer des contraintes sémantiques sur les structures formées et obtenir une représentation sémantique de la question. La méthode que nous proposons consiste à engendrer toutes les structures possibles pour représenter la question à partir de son graphe de dépendances syntaxiques. Ce graphe syntaxique est transformé en utilisant les relations, entités et types identifiés préalablement pour produire un ensemble de graphes sémantiques candidats, constitués de triplets bien formés. Ces triplets sont pondérés grâce aux scores déterminés lors de l'identification des éléments sémantiques, et permettent de pondérer les graphes qui peuvent être transformés en requête SPARQL afin d'obtenir une réponse.

La méthode d'analyse que nous proposons ne pose que des contraintes très générales concernant la structure de la requête et peut être appliquée dans différents contextes sémantiques. Par ailleurs, nous repoussons la résolution des ambiguïtés au plus tard, afin de pouvoir tenir compte de l'ensemble de la signification de la question. Nous évaluons notre méthode d'identification de relations sur des questions de QALD3 et obtenons de bons résultats. Afin d'évaluer les représentations construites nous évaluons aussi l'ensemble du système, et obtenons des résultats prometteurs.

2. Définitions

Une *relation* R associe des entités d'un domaine $D1$ à des entités d'un domaine $D2$, une *instance* de relation associe une entité $e1$ à une entité $e2$ par une relation R et une *mention* de relation est la réalisation phrastique d'une instance de relation dans un document. Par exemple *auteur_de(Personne,Œuvre)* est une relation, *auteur_de(Daniel_Defoe,Robinson_Crusoé)* est une instance de la relation et *Daniel Defoe est l'auteur de Robinson Crusoé* est une mention de la relation.

Les relations sont liées à la base de connaissance utilisée pour répondre aux questions. Une base de connaissance est en effet constituée d'entités reliées entre elles par des relations binaires ou n-aires. Ces bases de connaissance ne sont pas toutes structurées de la même façon : DBpedia, YAGO et MusicBrainz contiennent des triplets (donc des relations binaires) alors que Freebase contient des n-uplets (et donc des relations n-aires). Il est néanmoins possible de passer d'une représentation à l'autre.

Le langage standard d'interrogation de ces bases de connaissances est SPARQL, langage proche du SQL, qui permet d'interroger les données RDF des ressources du Web sémantique.

3. Présentation de l'approche proposée

L'approche que nous proposons consiste à transformer une question en un ensemble de représentations sémantiques sous forme de graphes G_Q^S (Graphe sémantique de la question) qui permettent d'interroger une base de connaissance. Chaque graphe représente une façon d'interpréter la question. Les noeuds du graphe sont des entités et les arêtes sont des relations. Chaque entité et relation est associée à un score de pertinence, qui permet de calculer le score du graphe.

Lors de l'analyse de la question des syntagmes sont identifiés afin de les relier à des éléments sémantiques de la base de connaissance : entités, types ou relations (figure 1a). L'identification des entités est réalisée par DBpedia spotlight (Daiber *et al.*, 2013), l'identification des types par leurs labels, et l'identification des relations par la méthode que nous proposons fondée sur des variations extraites de WordNet. À chaque possibilité trouvée correspond un score de pertinence.

L'analyse syntaxique de la question par le Stanford Parser (Klein et Manning, 2003) produit un graphe syntaxique qui est simplifié et réécrit pour former un graphe G_Q^{Sy} (Graphe syntaxique de Question)) ayant pour noeuds les mots et pour arêtes les relations syntaxiques. Dans ce graphe, la nature sémantique de chaque mot (entité ou relation) n'est pas connue (figure 1b). Ces graphes sont démultipliés en supposant que chaque mot peut être entité ou relation afin de conserver les ambiguïtés possibles, puis transformés et filtrés afin d'obtenir les graphes G_Q^{SS} qui respectent des contraintes structurelles de bonne formation (figure 1c). Par exemple, si deux $mot\text{-}relation$ sont liés par une relation syntaxique, cela signifie qu'il y a une entité implicite et on ajoute un $mot\text{-}entité$ entre eux. Ainsi, dans la question *Who is the daughter of Bill Clinton married to ?*, *daughter* et *married* sont des $mot\text{-}relation$ et sont reliés directement, on crée donc un $mot\text{-}entité$ inconnu entre les deux, qui explicite le fait de devoir trouver le nom de la fille de Bill Clinton pour répondre à la question.

Il reste ensuite à engendrer les graphes sémantiques possibles en fonction des ambiguïtés liées à chaque noeud ou arête obtenus après la phase d'appariement. Ces graphes sont donc constitués de triplets consistants avec la base de connaissance (figure 1d). Ces graphes sémantiques sont ensuite ordonnés selon leur score de pertinence et les requêtes correspondantes sont exécutées dans l'ordre jusqu'à trouver au moins une réponse.

La méthode que nous proposons vise à ne jamais éliminer des interprétations tant qu'elles sont cohérentes avec une structure sémantique globale. Les contraintes liées à la structure sémantique sont très générales, et non spécifiques au schéma relationnel d'une base de connaissances particulière ou à des patrons de requêtes liés au langage d'interrogation. De ce fait, elle peut être aussi utilisée pour rechercher des informations dans les textes. Les contraintes liées à la base interrogée sont appliquées en dernière étape et l'ultime filtrage est opéré par la recherche dans la base de connaissance, selon que l'on trouve ou non une réponse.

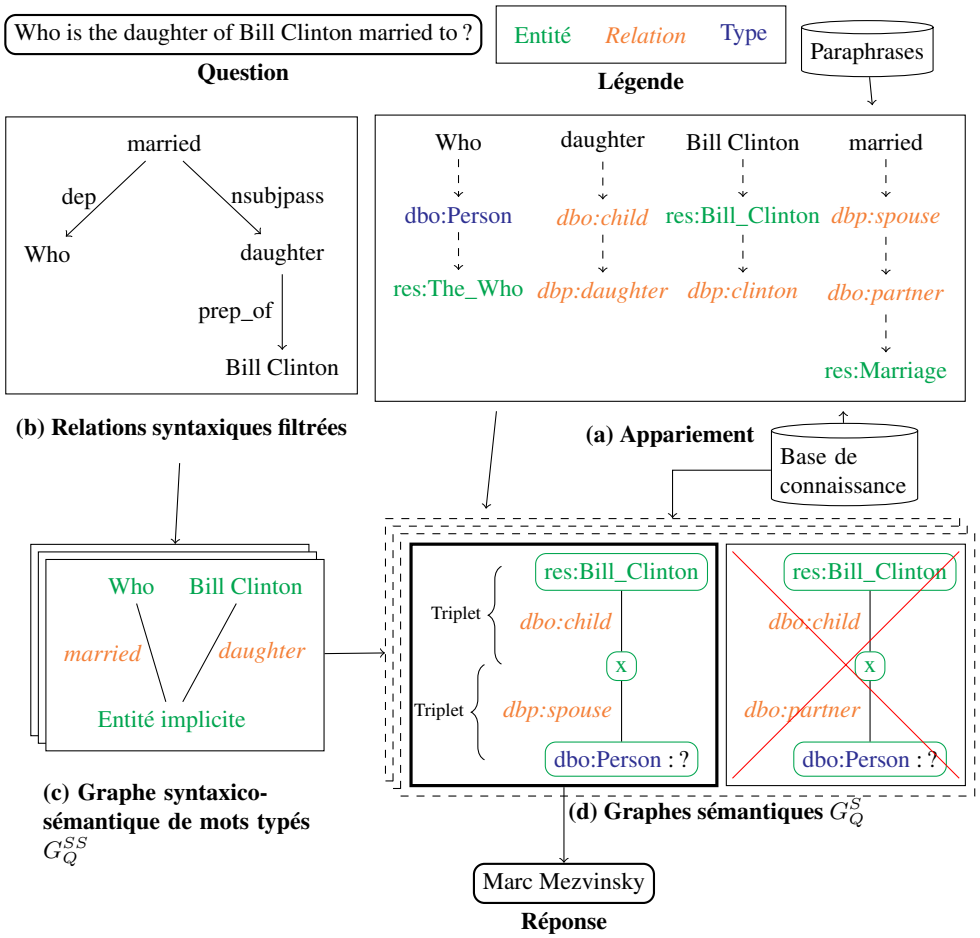

Figure 1 – Approche proposée

4. Appariement de termes à la base de connaissance

Afin d'apparier les termes de la question à des éléments sémantiques de la base de connaissance (entité, relation et type : voir figure 1a), nous déterminons un ensemble de syntagmes que nous chercherons à relier à ces éléments sémantiques. Cet ensemble de syntagmes est constitué de tous les n-grams qui contiennent un adjectif, nom ou verbe.

4.1. *Extraction et identification d'entités*

L'analyse des entités consiste à extraire leur mention, c'est-à-dire à détecter les mots ou syntagmes qui font référence à une entité, à les identifier, c'est-à-dire relier ces mots à une ressource d'une base de connaissance et enfin à détecter leur type.

DBpedia Spotlight (Daiber *et al.*, 2013) permet d'extraire, identifier et typer les entités d'un texte, c'est-à-dire d'extraire et lier une entité du texte avec une ressource DBpedia (représentée par une URI). DBpedia Spotlight classifie les entités en utilisant l'ontologie de DBpedia qui contient une hiérarchie de types [1].

Nous avons évalué l'annotation des entités DBpedia par Spotlight sur les questions de QALD3 et nous obtenons un rappel de 0,86 d'entités correctement reliées à DBpedia. C'est la valeur de rappel qui nous intéresse ici car l'objectif est d'identifier toutes les entités possibles, même si certaines semblent moins pertinentes, car en réalité elles peuvent être utilisées dans la suite du système. Nous avons donc choisi de conserver toutes les ambiguïtés.

Le score de confiance d'une entité est le score donné par DBpedia Spotlight, qui est calculé à partir de scores initiaux déterminés lors de l'indexation des entités et de scores contextuels calculés par rapport aux autres termes de la question.

4.2. *Identification des types*

Les questions auxquelles nous nous intéressons contiennent des syntagmes qui peuvent être associés à des types dans une base de connaissance. Par exemple dans *Which languages are spoken in Estonia ?*, *languages* correspond à *dbo:Language*. L'identification de ces types est une information importante pour répondre à ces questions. Les types que nous identifions sont ceux de l'ontologie de DBpedia mais aussi ceux de Yago[2]. Nous réalisons cette identification en comparant les syntagmes candidats de la question avec les étiquettes des types. Cela permet d'obtenir un rappel de 0,66 sur QALD3. Une fois ces types identifiés ils sont utilisés sous la forme d'entités inconnues qui ont ces types et qui seront instanciées seulement lors de la résolution de la requête SPARQL. Par exemple dans la question *Which software has been developed by organizations founded in California ?* , *organizations* correspond au type *dbo:Company* et *software* correspond au type *dbo:Software*

Les scores des types identifiés sont déterminés par la longueur en nombre de mots du type : plus le type est long plus il est spécifique, par exemple on préférera *StatesOfTheUnitedStates* à *States*. Le score utilisé pour évaluer la pertinence d'un type est donc :

$$Stype = \frac{nombre\ mots}{nombre\ mots\ maximal} \quad [1]$$

1. http ://mappings.dbpedia.org/server/ontology/classes/
2. DBpedia inclut les types de Yago (Hoffart *et al.*, 2011)

Nous avons majoré le nombre de mots à 10. Un deuxième facteur que nous considérons est que les types de l'ontologie de DBpedia sont moins nombreux (529 contre 379900 dans Yago) et plus présents dans les triplets de DBpedia. Ils sont donc préférés et nous avons donc multiplié le score par un coefficient α inférieur à 1 pour les types de Yago. Nous avons pris $\alpha = 0.5$.

4.3. *Extraction de relations*

L'extraction de relations a pour objectif de détecter les mentions de relation dans un texte, qui pourront ensuite être reliées à une relation sémantique lors de l'identification des relations.

Différents outils existent, nous les avons évalué sur un corpus de questions. ReVerb (Fader *et al.*, 2011) a été conçu pour extraire des relations dans de gros corpus. Il se fonde sur une analyse syntaxique partielle afin d'extraire des triplets de la forme (entité,relation,entité) avec entités et relation sous forme de texte. Il peut aussi produire des triplets ayant une entité manquante comme *(?,is the daughter of,Bill Clinton)*. Une relation est extraite à partir des verbes et des mots qui lui sont proches et qui respectent des contraintes exprimées sous forme d'expression régulière sur les catégories grammaticales. Les arguments des relations sont extraits en cherchant les groupes nominaux les plus proches. ReVerb a été construit pour extraire des relations dans des phrases affirmatives et appliqué aux questions, il obtient de moins bonnes performances. Sa précision est assez élevée mais son rappel est bas. Or pour l'extraction de triplets dans les questions le rappel est très important car il est nécessaire de ne pas perdre d'information de la question. Sur 25 questions de développement, 31 relations sont extraites par ReVerb mais seulement 6 sont de qualité suffisante pour être utilisées pour l'interrogation de la base de connaissance.

RelEx (Fundel *et al.*, 2007), quant à lui, se fonde sur une analyse syntaxique des phrases pour extraire des relations plus sémantiques entre les mots. Les graphes syntaxiques sont réécrits afin d'extraire des relations comme *in(live, New_York)* [3]. Son emploi sur des questions a amené également à des pertes d'information et donc une diminution du rappel . En effet les réécritures employées sont surtout faites pour des phrases déclaratives, et ne fonctionnent pas toujours pour des questions.

Notre objectif étant d'identifier un maximum de relations dans une question, nous avons choisi de développer une méthode prenant en entrée toutes les mentions de relation disponibles dans la question plutôt que le résultat d'un filtrage préalable en relation/non relation. Nous considérons ainsi que tous les mots pleins (adjectif, nom, verbe) sont potentiellement des mentions de relation.

Afin de déterminer à quelles relations de la base de connaissance correspondent ces mots, donc identifier les relations, nous avons cherché à détecter des variantes de ces mentions de relation.

3. http ://wiki.opencog.org/w/Binary_relations

4.4. *Identification de relations et traitement des variantes*

L'identification de relation consiste à associer les mentions potentielles de relation à une relation d'une ontologie. Rappelons qu'à chaque relation de l'ontologie est associé un label, qui représente une dénomination dans les textes. Il s'agit donc de relier une expression trouvée dans une question à une expression de relation donnée. Plusieurs cas de variations peuvent se présenter :

1) La catégorie syntaxique du label et de la mention de relation sont différentes, par exemple dans *Who is Barack Obama married to ?*, le verbe *married* doit être relié à la relation dont le label est le substantif *spouse*.

2) Les mentions de relation présentent souvent une distance sémantique importante avec le label. Cela peut venir du fait que la mention de relation fait référence à un hyponyme[4], est un synonyme, ou une autre variation sémantique du label.

3) Les labels et mentions de relation peuvent contenir plusieurs mots, qui ont chacun leurs propres variations.

Pour gérer ces variations nous avons envisagé plusieurs méthodes afin de construire un dictionnaire de variations à partir de labels :

1) utiliser un corpus de paraphrases : les paraphrases permettent d'identifier des mentions de relation équivalentes ;

2) utiliser une base de données lexicale, WordNet en l'occurrence, particulièrement pertinente dans le cas où la mention de relation est formée d'un seul mot ;

4.4.1. *Paraphrases*

Nous avons utilisé une base de paraphrases afin d'obtenir une couverture large des variations. Nous avons utilisé la base de paraphrases PPDB((Ganitkevitch *et al.*, 2013)) qui ont été extraites automatiquement à partir de corpus textuels.

L'ensemble de paraphrases, constitué d'un ensemble de paires d'expressions, est parcouru. Si un des labels de relation est dans la première phrase mais pas dans la deuxième alors la deuxième phrase est retenue comme variation de cette relation. Une fois ces variations obtenues, elles sont utilisées afin d'identifier les relations à partir de leur mention : si les mots de la mention sont présents dans une des variations, alors la relation est la relation associée à ces mots. Une fois ces mentions identifiées, les triplets sont formés en les reliant aux entités connues (qui ont été déterminées dans la phase d'extraction des entités).

Nous avons évalué la performance d'extraction des relations sur notre jeu de développement, les questions de la tâche hybride de QALD4. Pour évaluer l'extraction de relations, nous avons extrait les relations des requêtes SPARQL proposées dans la référence et les avons comparées avec les relations produites par notre système en terme de rappel. Plusieurs relations sont possibles, donc correctes pour une même mention

4. un concept plus spécifique

de relation, lorsque l'on parle d'identification indépendamment du contexte global de la phrase. C'est seulement avec la formation des triplets et l'interrogation de la base de connaissance qu'il est possible de lever l'ambiguïté. Or ici la référence (les requêtes SPARQL proposées par QALD) ne contient que la relation qui a permis de répondre à la question. C'est pour cette raison que nous ne calculons pas de précision.

Nous obtenons un rappel de 0,48. Les difficultés d'identification des relations proviennent de mentions de relation qui ne sont pas reconnues, notamment quand le label initial de la relation est un nom alors que son expression dans la question est un verbe (par exemple *born* est exprimé dans DBpedia comme *birthDate*). Pour remédier à ces difficultés, nous avons utilisé WordNet.

4.4.2. *WordNet*

WordNet est une ressource lexicale qui associe des mots à des concepts - les synsets - contient des relations lexicales entre mots - les formes dérivées, les synonymes, les définitions sous forme d'énoncés - et des relations sémantiques entre concepts, notamment les relations hiérarchiques d'hyperonymie et d'hyponymie.

Pour acquérir des variations de mention de relations, nous avons utilisé les relations suivantes de WordNet :

– formes dérivées : *successor* $\leftrightarrow$ *succeed*

– synonymes : *author* $\leftrightarrow$ *writer*

– hyperonymes/hyponymes : *relative* $\leftrightarrow$ *sister*

En partant d'un label, on parcourt ces trois relations récursivement avec une profondeur maximale d fixée. Fixer la profondeur maximum permet de limiter la quantité de variations générées et de ne conserver que les meilleures. Nous avons fixé expérimentalement la profondeur maximale à $d = 4$.

Sur le jeu de test QALD3, nous obtenons un rappel de 0,68. Sur la tâche hybride de QALD4 nous obtenons un rappel de 0.92. La différence de score peut s'expliquer par le fait que les relations de la base de connaissance sont plus facile à identifier sur la tâche hybride de QALD4 que sur QALD3 car sur la tâche hybride une difficulté supplémentaire est de traiter des relations textuelles, ce qui n'est pas évalué ici.

Pour chaque mention de relation, plusieurs relations sont en général trouvées grâce aux différentes variations provenant de WordNet. Afin d'évaluer leur pertinence un score entre 0 et 1 est attribué à chaque relation candidate qui est déterminé par :

$$S_r = \frac{1}{longueur\ du\ chemin\ is-a\ reliant\ les\ deux\ termes} \qquad [2]$$

La longueur du chemin *is-a* correspond au nombre de lien d'hyperonymes/hyponymes traversés.

4.5. *Conclusion*

Nous n'avons retenu que la méthode fondée sur WordNet, ses résultats étant meilleurs. Nous avons construit un dictionnaire de variations pour l'ensemble des relations de DBpedia, qui ont toutes un label en anglais, par exemple *dbo:child* qui indique les enfants d'une personne, de label *child* ou *dbo:birthDate* qui indique la date de naissance d'une personne, de label *birth date*. Au total 1 252 327 variations sont générées pour 12 331 relations, avec une moyenne de 102 variations par relations.

5. Formation d'une représentation sémantique de la question

5.1. *Représentation sémantique*

Le sens d'une question est représentée sous la forme d'un ensemble de triplets (e_1, R, e_2), avec e_1 et e_2 des entités et R une relation, formant un graphe sémantique G_Q^S connexe et pondéré.

Une entité est décrite par quatre champs : *mention*, *type*, *valeur*, *score*. Les champs *mention* et *type* peuvent être vides. Le champ *valeur* correspond à une *uri*, une variable x lorsqu'il s'agit d'une entité implicite dans la question, ou ? lorsqu'il s'agit de l'entité à trouver. Le score provient du processus d'identification (cf section 4.1), et a une valeur quand on a trouvé l'*uri*. Par exemple un mot interrogatif est une entité avec un type et un label. Une entité implicite possédera seulement un label (généré).

Une relation possède un label (dans la base de connaissance), un domaine, un co-domaine et un identifiant dans la base de connaissance (uri). Par exemple la relation *dbo:birthDate* (*http ://dbpedia.org/ontology/birthDate*) a pour label *birth date*, pour domaine *dbo:Person* et pour co-domaine *xsd :date*

Les entités ne sont reliées par des relations que si une relation syntaxique est présente entre leurs mentions.

Ce type de représentation est engendré à l'issue de l'analyse des questions. Lorsqu'il y a des ambiguïtés des termes de la question par rapport aux éléments de la base de connaissance, on formera plusieurs graphes sémantiques.

5.2. *Formation des triplets*

Les triplets sont formés à partir des relations de dépendance syntaxique obtenues après application du Stanford parser (Klein et Manning, 2003). Ce processus suit deux grandes étapes : la production de graphes syntaxico-sémantiques, G_Q^{SS}, vérifiant des critères de bonne formation syntaxique des graphes sémantiques G_Q^S à produire, et ensuite la génération des graphes sémantiques G_Q^S qui respectent des contraintes sémantiques.

La transformation d'un graphe syntaxique en un graphe G_Q^{SS} suit les étapes ci-dessous :

a) le graphe syntaxique est simplifié pour ne conserver que les relations de dépendances pertinentes entre les mots de la question. Les dépendances sont filtrées suivant le type de la relation syntaxique afin de ne conserver que celles qui sont utiles pour cette tâche. Les relations syntaxiques determiner, noun compound modifier, controlling subject ,passive auxiliary, auxiliary, open clausal complement, controlling subject et copula ont été enlevées. Les relations restantes forment un graphe de mots (voir la figure 1b) ;

b) un mot peut dénoter une ou plusieurs entités ou une ou plusieurs relations. Cela entraîne des structures différentes de graphe. On génère donc les graphes correspondant à toutes les possibilités où chaque terme est remplacé par un $mot\text{-}entité$ ou un $mot\text{-}relation$. Si la mention, d'un $mot\text{-}entité$ ou d'un $mot\text{-}relation$ correspond à plusieurs entités ou relations de la base respectivement, on garde les ambiguïtés associées au noeud ;

c) la liste de ces graphes est ensuite filtrée par des critères de bonne formation : chaque $mot\text{-}entité$ doit être lié à un $mot\text{-}relation$; si deux $mot\text{-}relation$ sont reliées, on ajoute un $mot\text{-}entité$ correspondant à une entité implicite ; chaque $mot\text{-}relation$ doit être liée à deux $mot\text{-}entité$; le graphe doit être connexe, pour être certain que les triplets soient reliés entre eux (cf figure 1c) ;

Une fois ces graphes G_Q^{SS} de mots typés en relation ou entité formés, il est possible d'engendrer les graphes sémantiques G_Q^S formés de triplets sémantiques :

1) Un triplet est formé à partir de chaque sous-graphe liant deux entités par une relation. Lorsqu'il y a ambiguïté, on prend tous les candidats entités associés aux mots entités et les candidats relations associés aux mots relations pour former tous les triplets sémantiques possibles ;

2) On vérifie ensuite la cohérence sémantique des triplets, en conservant seulement les triplets qui permettent d'obtenir une réponse dans la base de connaissance ;

3) On calcule ensuite le produit cartésien entre ces listes de triplets sémantiques pour obtenir les graphes sémantiques G_Q^S, représentations sémantiques possibles de la question (cf figure 1d)

Le fait de passer par les graphes G_Q^{SS} avant d'engendrer les graphes G_Q^S, permet de filtrer une partie des graphes sur des critères de structure, sans avoir à tenir compte des ambiguïtés sur chaque noeud et arc, et produire ainsi moins de graphes sémantiques.

5.3. *Identification des triplets incompatibles*

Une fois tous les triplets possibles connus, on peut évaluer leur bonne formation par rapport aux contraintes sémantiques de la base.

Le rattachement des entités par les relations permet d'identifier quelles relations candidates ne sont pas compatibles avec les entités auxquelles elles sont rattachées. Ces relations candidates incompatibles se voient attribuer un score plus faible afin de n'être choisies pour la formation de requête que si aucun autre candidat ne permet d'obtenir une réponse. En effet on ne peut pas simplement les rejeter car la base de connaissance (et en particulier DBpedia) peut être incohérente : certaines entités sont reliées par une relation sans respecter le domaine ou le co-domaine. Par exemple la réponse à la question *Who developed Minecraft ?* est dans DBpedia sous la forme du triplet *res:Minecraft dbo:developer res:4J_Studios*. Le domaine de *dbo:developer* est *dbo:UnitOfWork* et le co-domaine est *dbo:Agent*. *res:Minecraft* a pour type *dbo:VideoGame* qui n'est pas compatible avec *dbo:UnitOfWork*.

Afin de tester la compatibilité entre deux entités et une relation il faut définir la compatibilité entre une entité e de type C_e et un domaine D de type B : e est considérée incompatible avec D seulement si B n'est pas inclus dans l'ensemble formé de C et de ses super types et sous types.

Il existe trois possibilités : compatibilité, incompatibilité et le cas où l'information est manquante si on ne connaît pas le domaine ou le type de l'entité.

Par exemple la relation *dbo:childOrganisation* identifiée comme relation candidate de la mention *daughter* et qui relie *Bill Clinton* et sa fille (mention d'entité implicite ici et dont le type est inconnu) est incompatible car elle a pour domaine d'arrivée *dbo:Organisation* qui n'est pas compatible avec le type *dbo:President* de *Bill Clinton*.

Ces informations de compatibilité permettent de déterminer un score Sc de compatibilité d'un triplet, calculé ainsi :

$$Sc = Scd \times Scr \quad [3]$$

avec Scd et Scr les scores de compatibilité au niveau du domaine et du co-domaine. Ils prennent pour valeur β en cas de compatibilité, γ en cas d'information manquante et δ en cas de non compatibilité. Un domaine compatible doit obtenir un meilleur score qu'en cas d'information manquante ou de non compatibilité donc on a $\beta > \gamma > \delta$ et nous avons fixé $\beta = 1.0$, $\gamma = 0.75$ et $\delta = 0.5$.

5.4. *Pondération d'un graphe sémantique*

A chaque graphe sémantique est associé un score qui permet de le pondérer. Les graphes sont ensuite convertis en requêtes et exécutés dans l'ordre décroissant de ces scores. Ce score S est donné par la formule :

$$S = \frac{\sum_{i=1}^{n} St_i}{n} \quad [4]$$

avec n le nombre de triplets dans le graphe, et St_i le score du triplet i

$$St_i = Sc\frac{Se_1 + Sr + Se_2}{3} \quad [5]$$

avec Se_1 et Se_2 les scores des deux entités, Sr le score de la relation et Sc le score de compatibilité du triplet.

6. Expérimentations et résultats

6.1. *Ressources*

La base interrogée DBpedia (Lehmann *et al.*, 2014) est une base de connaissance contenant de nombreuses entités reliées par des relations binaires qui ont été extraites automatiquement des infobox de Wikipédia (encadré formaté contenant des informations importantes). Afin d'améliorer la qualité de cette base, une ontologie a été construite et ces entités et relations y ont été reliées. Il existe deux types de relations : les relations brutes issues des infobox et les relations de l'ontologie. Les relations brutes sont présentes en plus grand nombre mais sont souvent de mauvaise qualité (ambiguïté, label qui n'a pas de sens) alors que les relations de l'ontologie sont non ambiguës et de meilleure qualité. Les relations de l'ontologie ont été construites à partir des relations des infobox : plusieurs relations des infobox ayant un nom proche et ayant le même sens peuvent être reliées à une seule relation de l'ontologie. En revanche, elles ne sont pas disponibles pour toutes les entités (les associations entre relations brutes et relations de l'ontologie étant construites manuellement). Chaque relation possède un label qui est utilisé pour identifier les relations dans les questions. Les triplets (entité, relation, entité) sont représentés au format RDF et peuvent être interrogés avec le langage de requête SPARQL.

Les évaluations se font sur le corpus de questions de la première tâche de QALD3 [5]. Celui-ci est constitué de 99 questions en anglais qu'il faut traduire en SPARQL afin d'interroger DBpedia. A ces 99 questions sont associées les traductions en SPARQL ainsi que les réponses. Les requêtes SPARQL proposées par la référence contiennent des relations de typage des entités, des relations textuelles et des relations contenues dans DBpedia.

6.2. *Résultats*

L'objectif des expérimentations est d'évaluer le bien fondé d'utiliser les variations sémantiques afin d'identifier les relations et d'évaluer la construction de la représentation sémantique. Pour cela nous avons évalué les réponses obtenues dans deux cas : sans utiliser de variations sémantiques sur les variations et en les utilisant. Les graphes sémantiques sont appliqués dans l'ordre et quand une réponse est obtenue elle est conservée.

5. http ://greententacle.techfak.uni-bielefeld.de/ cunger/qald/index.php ?x=task1&q=3

Système	#Questions	#Répondues	#Répondues correctement	Rappel	Précision	F-mesure
Sans variations	99	22	16	0,15	0,16	0,15
Complet	99	45	28	0,28	0,26	0,26
CASIA	99	52	37	0,36	0,35	0,36
Moyenne	59	52	31	0,30	0,30	0,30

Tableau 1 – Résultats de l'évaluation sur QALD3

Dans le tableau 1 sont présents les résultats de l'évaluation du système sans prendre en compte les variations sémantiques sur les relations et en les prenant en compte (complet). On peut observer que ces variations permettent d'améliorer de façon importante les résultats obtenus (0,26 de f-mesure contre 0,15). Nos résultats sont proches de la moyenne des résultats obtenus à QALD3 (0,26 contre 0,30) et du meilleur système automatique CASIA (He *et al.*, 2013) (0,26 contre 0,36).

Parmi les questions qui n'obtiennent pas de réponses correctes, 20% possèdent des entités qui ne sont pas reconnues, 45% des relations non reconnues et 15% des type non reconnus. Cela peut s'expliquer par le fait que certaines relations ne sont pas lexicalisées (par exemple dans *Give me all movies with Tom Cruise.* il faut reconnaître *dbo:starring*) ou bien qu'elles sont difficiles à reconnaître (par exemple dans *Give me all cars that are produced in Germany.*, *produced* doit être associé à *dbp:assembly*).

Il y a néanmoins 43% de questions qui n'ont pas obtenu de réponse pour lesquelles tous les relations, entités et types sont trouvés. Pour ces questions les erreurs peuvent provenir de problèmes concernant la gestion des ambiguïtés, la détection de la structure de la question ou bien la nécessité de faire des agrégations (*What is the second highest mountain on Earth ?* par exemple).

7. État de l'art

L'interrogation de bases de connaissance en langage naturel a bénéficié des avancées dans le domaine des questions-réponses dans des textes, de l'interface en langage naturel pour les bases de données ou de connaissances, et de la recherche dans le web sémantique. Nous présenterons ici principalement les travaux sur la réponse aux questions dans des bases de connaissance, par rapport auxquels nous nous positionnerons.

De nombreux travaux ont été réalisés récemment sur la tâche de question-réponse sur les bases de connaissances. (Fader *et al.*, 2013) propose d'utiliser des patrons de questions et d'apprendre à les associer à leur formulation sous forme de question. Par exemple *how r is e = r(?, e)* sera associé à la question *how big is nyc ?* et la transformera (grâce à un lexique d'entités et de relations) en *population(?,new-york)*. Mais cette approche ne traite que les questions composées d'un seul triplet. Pour des questions complexes le nombre de variations devient trop important et il est préférable

d'étudier les variations des mentions de relations plutôt que de la question entière. (He *et al.*, 2013) traite les ambiguïtés à priori, et utilise des patrons de question afin de déterminer le type de requête approprié.

Afin de gérer les ambiguïtés présentes dans la question et construire une représentation de celle-ci, (Yahya *et al.*, 2013) propose d'utiliser la programmation linéaire en nombre entier (ILP) et (He *et al.*, 2014) utilise les réseaux logiques de Markov (MLN). Un choix est réalisée avant l'interrogation de la base, alors que nous avons choisi de repousser ce choix au plus tard et de l'effectuer lors de l'interrogation de la base, de manière analogue à (Zou *et al.*, 2014). Dans ce dernier travail, la construction de représentations sémantiques est dirigée par la base DBpedia, alors que nous proposons une méthode générique, fondée sur les relations syntaxiques. De plus, la plupart de ces travaux ne se concentrent pas sur l'identification des variations sémantiques des relations et utilisent la ressource Patty. Nous avons réalisé une étude plus spécifique de ces variations et avons utilisé WordNet.

8. Conclusion

Dans le cadre de systèmes permettant de répondre à des questions sur des bases de connaissances du Web sémantique, nous avons proposé des méthodes permettant de répondre aux problèmes d'identification des relations et de traitement des ambiguïtés. Afin d'identifier les relations, nous avons intégré le traitement des variations sémantiques provenant de WordNet afin de construire un dictionnaire de variantes de labels de relations. Pour le traitement des ambiguïtés structurelles et lexicales, nous avons proposé une méthode se basant sur des transformations de graphe. Les résultats obtenus sont proches des résultats de l'état de l'art sur la tâche QALD3. Les méthodes que nous avons utilisées sont indépendantes de la base de connaissance et pourront s'appliquer à d'autres bases que DBpedia. La représentation sémantique formée est indépendante du langage de requête. La prochaine étape sera l'intégration d'une méthode plus performante pour choisir la requête ou la réponse parmi les graphes sémantiques construits.

Par la suite, nous prévoyons de construire une interrogation hybride de bases de connaissance et de textes en utilisant la représentation sémantique formée et en adaptant l'identification d'éléments sémantiques afin de prendre en compte des informations présentes dans des corpus textuels.

9. Bibliographie

Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z., *Dbpedia : A nucleus for a web of open data*, Springer, 2007.

Bollacker K., Evans C., Paritosh P., Sturge T., Taylor J., « Freebase : a collaboratively created graph database for structuring human knowledge », *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ACM, p. 1247-1250, 2008.

Daiber J., Jakob M., Hokamp C., Mendes P. N., « Improving Efficiency and Accuracy in Multilingual Entity Extraction », *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

Fader A., Soderland S., Etzioni O., « Identifying Relations for Open Information Extraction », *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, Edinburgh, Scotland, UK, July 27-31, 2011.

Fader A., Zettlemoyer L., Etzioni O., « Paraphrase-driven learning for open question answering », *Association for Computational Linguistics (ACL)*, 2013.

Fundel K., Küffner R., Zimmer R., « RelEx—Relation extraction using dependency parse trees », *Bioinformatics*, vol. 23, nº 3, p. 365-371, 2007.

Ganitkevitch J., Van Durme B., Callison-Burch C., « PPDB : The Paraphrase Database », *Proceedings of NAACL-HLT*, Association for Computational Linguistics, Atlanta, Georgia, p. 758-764, June, 2013.

He S., Liu K., Zhang Y., Xu L., Zhao J., « Question Answering over Linked Data Using First-order Logic », *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

He S., Liu S., Chen Y., Zhou G., Liu K., Zhao J., « Casia@ qald-3 : A question answering system over linked data », *Proceedings of the Question Answering over Linked Data lab (QALD-3) at CLEF*, 2013.

Hoffart J., Suchanek F. M., Berberich K., Lewis-Kelham E., De Melo G., Weikum G., « YAGO2 : exploring and querying world knowledge in time, space, context, and many languages », *Proceedings of the 20th international conference companion on World wide web*, ACM, p. 229-232, 2011.

Klein D., Manning C. D., « Accurate unlexicalized parsing », *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, Association for Computational Linguistics, p. 423-430, 2003.

Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., Mendes P. N., Hellmann S., Morsey M., van Kleef P., Auer S. *et al.*, « DBpedia–A large-scale, multilingual knowledge base extracted from Wikipedia », *Semantic Web*, 2014.

Nakashole N., Weikum G., Suchanek F., « PATTY : a taxonomy of relational patterns with semantic types », *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, p. 1135-1145, 2012.

Unger C., Bühmann L., Lehmann J., Ngonga Ngomo A.-C., Gerber D., Cimiano P., « Template-based question answering over RDF data », *Proceedings of the 21st international conference on World Wide Web*, ACM, p. 639-648, 2012.

Yahya M., Berberich K., Elbassuoni S., Weikum G., « Robust question answering over the web of linked data », *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, ACM, p. 1107-1116, 2013.

Zou L., Huang R., Wang H., Yu J. X., He W., Zhao D., « Natural Language Question Answering over RDF : A Graph Data Driven Approach », *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, ACM, New York, NY, USA, p. 313-324, 2014.