Approche hybride de questions-réponses basée sur le traitement automatique des langues et les requêtes SPARQL

Mickael Rajosoa*, Rim Hantach*, Sarra Ben Abbès*, Philippe Calvez*

*LAB CSAI ENGIE, 361 Avenue du Président Wilson, 93210 Saint-Denis, France
Rajosoa.Mickael@gmail.com, Rim.Hantach@external.engie.com
Sarra.Ben-Abbes@external.engie.com, Philippe.Calvez1@engie.com

Résumé. Le chatbot est un agent conversationnel qui communique avec les utilisateurs en langage naturel. Il est fondé sur un système de questions-réponses, les questions traitant l'intention de l'utilisateur. Dans ce contexte, des travaux récents ont été abordés présentant certaines limites. L'originalité de notre approche consiste à combiner les méthodes de traitement automatique du langage naturel avec les techniques du web sémantique. Une ontologie de domaine sert de base de connaissances pour décrire les informations dans un triplestore RDF. Les premiers résultats expérimentaux montrent l'intérêt de nos propositions.

1 Introduction

De nos jours, on est face à un grand volume de données, ce qui rend la recherche d'informations plus difficile. Pour surmonter ce problème, plusieurs approches de Questions-Réponses (QR) (Barskar et al., 2012; Yao et Durme, 2014) ont été proposées permettant de comprendre les questions en langage naturel et d'en extraire les informations pertinentes. QR est une discipline informatique conçue pour répondre d'une manière adéquate à des questions posées par les utilisateurs. Ce système est fondé sur (1) les méthodes de traitement automatique du langage naturel (TALN), pour analyser la question, et (2) la recherche d'information (RI), pour apporter une réponse adaptée à l'aide des documents qu'il possède (Chiticariu et al., 2013; Ferré, 2016). Afin de structurer les connaissances et de faciliter l'accès à ces documents, les techniques du Web Sémantique (WS) sont nécessaires. L'ontologie est employée en WS comme étant un modèle consensuel définissant les notions-clés (concepts et relations) d'un domaine spécifique et permettant de raisonner sur ces connaissances. Cependant, les travaux récents n'ont pas beaucoup développé ces techniques. Ils s'appuient sur des mots-clés pour identifier le contexte et les réponses aux questions de l'utilisateur. Ces méthodes se basent sur la suppression des mots vides impactant le sens de la phrase. Notre travail s'inscrit dans le cadre de l'amélioration des travaux de l'état de l'art et cela en combinant les méthodes de TALN et les techniques du Web Sémantique. Notre objectif principal est d'identifier l'intention de l'utilisateur à partir d'une analyse linguistique et sémantique de sa question. Dans cet article, nous passons en revue les travaux connexes dans la section 2. Dans la section 3, nous décrivons notre approche qui se repose sur l'utilisation des relations syntaxiques pour comprendre la question de l'utilisateur et apporter une réponse sémantique en utilisant des règles prédéfinies. Nous menons une étude

comparative dans la section 4 afin de démontrer l'efficacité de notre approche. Dans la section 5, nous présentons la conclusion et les perspectives.

2 État de l'art

Plusieurs approches de chatbot ont été abordées dans l'état de l'art où la plupart d'entre elles sont basées sur la préparation en amont d'un modèle de questions-réponses. En effet, Arain et al. (2018) décrit un agent conversationnel guidé et contrôlé par des patrons prédéfinis de questions. Quand l'utilisateur pose une question et qu'elle est dans la liste des questions prédéfinies alors le bot peut fournir une réponse adéquate. Cependant, cette approche a révélé un certain nombre de problèmes et limites liés à la fiabilité des réponses. En pratique, il est impossible d'énumérer toutes les questions qu'un utilisateur peut poser. Par conséquent, si la question d'un utilisateur n'est pas présente dans la base de questions, le bot ne peut pas fournir une réponse. Bansal et Chawla (2014) ont proposé un outil nommé Quepy pour transformer la question de l'utilisateur en requête SPARQL afin de trouver la réponse dans les "données ouvertes liées" telles que DBpedia [1]. Les auteurs mettent l'accent sur la reconnaissance des entités nommées (ex. personne, lieux, organisations, etc.) et l'identification des modèles de requêtes sous forme d'expressions régulières afin de générer la requête SPARQL. Néanmoins, une telle approche est vouée à l'échec puisqu'elle est fondée sur un modèle préparé. Ainsi, on se retrouve avec les mêmes limites que la première approche. Pour remédier à cela, Kulkarni et al. (2017) proposent une nouvelle approche basée sur les techniques d'apprentissage et de TALN. La méthode comporte 4 étapes : (1) préparation des questions possibles pour entraîner le bot, (2) utilisation des outils de TALN pour vectoriser la question, (3) application d'une mesure de similarité pour retrouver la question qui se rapproche le plus à la question posée, et (4) classification de la question selon les catégories prédéfinies.Cette approche est limitée car il est impossible d'évaluer si la liste de question est représentative du domaine d'étude. De surcroît, l'apprentissage automatique peut amener des erreurs car l'utilisateur peut employer des synonymes dans sa question. De ce fait, l'agent conversationnel ne peut pas répondre correctement à la question.
Dans (Bouziane et al., 2018), une nouvelle approche a été proposée, basée sur la combinaison de méthodes sémantiques et de TALN pour améliorer la capacité du chatbot. En donnant une question, des mots-clés et des entités nommées ont été extraits. Par conséquent, les mots-clés expriment l'intention de la question et aident à la construction de la requête SPARQL. Néanmoins, supprimer les mots vides et n'extraire que les mots-clés afin d'identifier l'intention de la question, peuvent limiter sévèrement son efficacité et induire des réponses erronées. Dans (Albarghothi et al., 2017), les chercheurs combinent aussi bien une approche linguistique et les techniques du web sémantique. Ils utilisent les fonctions de TALN pour traduire le langage naturel en requête SPARQL et interroger une ontologie. Dans (Ngonga Ngomo et al., 2013), une nouvelle approche a été établie pour transformer une requête SPARQL en langage naturel. Pour ce faire, différentes règles (ces règles ressemblent à des motifs) ont été définies pour construire les phrases. Malheureusement, cette approche souffre des aspects syntaxiques et sémantiques. En fait, les phrases générées peuvent être mal accordées en raison de la superposition de règles. Ainsi, les nouvelles phrases générées manquent du sens ou sans règles

1. https ://wiki.dbpedia.org/

grammaticales correctes.
Dans le travail de (Pradel et al., 2014), l'approche proposée repose sur les étapes suivantes : (1) interprétation de la requête de l'utilisateur en se basant sur l'analyseur syntaxique de dépendances MaltParser [2] pour construire une requête pivot, et (2) génération des patrons de requêtes qui seront proposés à l'utilisateur selon leur pertinence. L'outil proposé reste manuel et nécessite l'intervention d'un expert de domaine pour qualifier la pertinence des mots de la requête de l'utilisateur. Différentes approches de l'état de l'art ignorent les relations sémantiques et syntaxiques pour comprendre l'intention de la question. Notre travail (Rajosoa et al., 2019) s'inscrit dans ce cadre. Il présente la particularité d'automatiser l'analyse sémantique de la phrase d'un utilisateur, qui sera traduite sous forme de requêtes SPARQL en se basant sur un ensemble de règles.

3 Approche proposée fondée sur des règles linguistiques et sémantiques

Notre approche est basée sur la combinaison des méthodes de TALN et des techniques du web sémantique. Le but de cette combinaison est de comprendre la question de l'utilisateur. Comprendre signifie obtenir l'intention de l'utilisateur (que cherche-t-il derrière sa question ?) afin de fournir une réponse correcte. Notre approche exige une ontologie pour représenter l'information et les relations liées aux sujets. L'étape principale est d'analyser les mots d'une phrase. Il faut souligner qu'une partie de mots d'un discours ne suffit pas pour comprendre le sens d'une phrase. Il est également nécessaire de traiter la fonction syntaxique des différents mots et de détecter les entités nommées qui peuvent nous aider à percevoir le sens de la question et à discerner l'intention. Ainsi, nous utilisons les outils de TALN pour extraire la structure syntaxique de la phrase. Ensuite, cette information linguistique sera utilisée pour construire les règles. Enfin, ces règles seront transformées en requêtes SPARQL pour interroger le triplestore. Des ressources externes telles que WordNet (Miller, 1995) et DBpedia (Auer et al., 2007) ont été utilisées pour améliorer et renforcer la fiabilité de notre chatbot.
Dans le reste de cette section, nous détaillons ces différentes étapes.

3.1 Traitement des requêtes

Cette étape a pour but d'analyser la structure linguistique et syntaxique de la question : quel est le sujet de la question ? Quel le verbe principal ? Quels sont les compléments (objet, nom) ? La phrase comporte-t-elle une caractéristique particulière (e.g. coordination de conjonctions, une question avec une copule) ?
Nous utilisons l'analyse morphosyntaxique pour le sujet de la phrase, le noyau (verbe principal), ainsi que ses satellites (catégories simples ou groupes). L'outil de TALN permet de nous identifier les entités nommées ainsi que les relations de dépendances pour déterminer les fonctions syntaxiques et les relations entre les mots. Le tableau 1 montre les relations de dépendances de l'outil utilisé Stanford CoreNLP [3].
Nous prenons l'exemple suivant : Quel est le travail de Pierre ?

2. http ://www.maltparser.org/
3. https ://stanfordnlp.github.io/CoreNLP/

Annotations de Stanford CoreNLP	Fonction syntaxique
ROOT	prédicat principal
nsubj	sujet nominal / sujet
nmod	complément nominal
cop	copule
cc	coordination
conj	conjonction
advmod	adverbe

TAB. 1 – *Dépendances syntaxiques*

Dépendance syntaxique = *[('ROOT', 0, 1), ('cop', 1, 2), ('det', 4, 3), ('nsubj', 1, 4), ('case', 6, 5), ('nmod', 4, 6), ('punct', 1, 7)]*

3.2 Formalisation des règles

Au cours de cette étape, nous développons un modèle de règles généralisées basé sur les relations syntaxiques dans une question. Les règles définies peuvent être adaptées à n'importe quelle question (qui, quoi, où, quand, comment, etc.). Elles sont présentées comme suit :
Règle 1 : quand la question contient un complément du nom. Cela, signifie que le complément donne une information supplémentaire à sa tête syntaxique. Ainsi, cette tête syntaxique devient le prédicat secondaire.
Exemple : Comment s'appelle le mari de X ?
s'appelle : est le prédicat principal de la question.
mari : est la tête syntaxique du complément « de X », c'est donc le prédicat secondaire de la question.
X : est une entité nommée de type Person.
Règle 2 : quand la question contient des conjonctions de coordination tel que *et*, *ou*. Cela signifie que les entités nommées se trouvent sur le même niveau. En d'autres termes, ils partagent le même prédicat principal.
Exemple : qui sont les filles de X et Y ?
filles : est le prédicat principal de la question.
X et Y : sont des entités nommées de type Person.
Règle 3 : la question contient un prédicat non verbal comme par exemple une copule (la plus fréquente étant le verbe être) et que cette copule est précédée par le prédicat principal (qui est lui de type adverbe interrogative). Alors, on considère que le sujet de la question symbolise l'intention de l'utilisateur.
Exemple : quel est le travail de X ?
quel : est le prédicat principal de la question.
est : est le copule de la question.
travail : est le sujet de la question.
X : est une entité nommée de type Person.
Règle 4 : Les autre questions simples qui ont un sujet, prédicat principal et un complément.
Exemple : où se trouve X ?
trouve : est le prédicat principal de la question.

X : est une entité nommée de type Location.
Ces règles peuvent parfaitement se combiner entre elles, c'est-à-dire, que si une question suit une règle, elle n'exclut pas les autres règles. Par exemple, nous appliquons les règles 1, 2 et 3 pour cette question : "Comment s'appelle le mari de **S**, qui a les filles **X** et **Y** et qui travaille à **Z** ?"
s'appelle : est le prédicat principal de la question.
mari : est la tête syntaxique de S.
S : est une entité nommée de type Person.
qui : est le modifieur de la question.
X et Y : sont des entités nommées de type Person.
travaille : est en relation de dépendance avec mari.
Z : est une entité nommée de type Location.
La question principale des dépendances syntaxiques est l'identification de l'intention de l'utilisateur. En effet, il arrive parfois que les méthodes de TALN ne permettent pas d'identifier correctement l'intention. C'est pourquoi des ressources externes (DBpedia, WordNet) ont été utilisées pour traiter les relations sémantiques.

3.3 Construction des requêtes SPARQL

Avant de construire les requêtes SPARQL associées aux règles mentionnées dans la section précédente, nous avons construit une ontologie de domaine. Une ontologie est une base de données orientée graphe, structurée de façon hiérarchique et sémantique. Nous définissons pour chacune des règles génériques une requête SPARQL. En effet, le "ROOT" est le sommet de la question (prédicat principal) et la première propriété qui va relier une ressource A à une ressource B. Ensuite, si on possède des modifieurs ou compléments, alors leurs têtes se traduisent par une deuxième propriété (prédicat secondaire) qui va associer une ressource C à une ressource A. Quant aux conjonctions de coordination, nous faisons appel à la fonction UNION et si le "ROOT" est un adverbe interrogatif alors nous considérons que le prédicat principal est le sujet nominal de la phrase.
Les requêtes SPARQL pour les différentes règles sont générées automatiquement et définies comme suit :

Règle 1

```
SELECT DISTINCT ?a
WHERE
{?ans onto:main_predicate ?a
?x onto:secondary_predicate ?ans}
```

Règle 2

```
SELECT DISTINCT ?ans_label
WHERE
     {?x onto:main_predicate ?ans
      ?y onto:main_predicate ?ans
      ?ans rdfs:label ?ans_label.
     {?x rdf:type onto:Person
      ?x rdfs:label 'X'.}
UNION
     {?y rdf:type onto:Person
      ?y rdfs:label 'Y'.}}
```

Règle 3

```
SELECT DISTINCT ?ans_label
WHERE
     {?x onto:nominal_subject ?ans
      ?ans rdfs:label ?ans_label.}
```

Règle 4

```
SELECT DISTINCT ?a
WHERE
     {?ans onto:main_predicate ?a}
```

3.4 Interrogation du triplestore

Cette étape permet de répondre à l'intention de l'utilisateur. Étant donné qu'on a transformé les règles en requête SPARQL, nous interrogeons le triple store via SPARQL-Wrapper en utilisant le SPARQL endpoint (voir FIG. 1). Nous avons eu aussi recours à des ressources externes pour réduire les ambiguïtés sémantiques ; l'utilisateur peut utiliser des termes spécifiques dans sa question où les outils TAL ne sont pas en mesure de les identifier. Ce problème est résolu en utilisant : (i) *WordNet* qui réduit la polysémie des mots, et (ii) *DBpedia* qui permet d'identifier le type de chaque entité nommée.

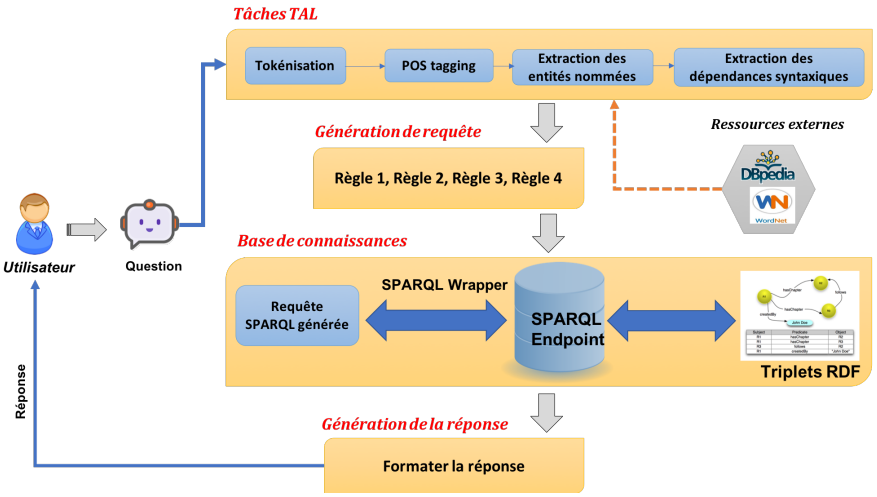

FIG. 1 – *Architecture du chatbot intelligent*

4 Évaluation

Pour évaluer la performance de notre approche, nous présentons une expérimentation comportant un jeu de test de questions-réponses et une ontologie de domaine. Nous avons utilisé le même corpus de Bouziane et al. (2018) composé de 193 questions/réponses et l'ontologie Person (11 classes, 33 propriétés et 86 instances) qui décrit les informations d'une personne à savoir sa date de naissance, son lieu de naissance, sa profession, etc. Nous avons mis à jour cette ontologie de base avec des relations sémantiques. L'objectif principal est d'évaluer la capacité du chatbot à identifier l'intention de l'utilisateur.
Afin de comparer notre travail avec celui de Bouziane et al. (2018), nous avons utilisé les mesures d'évaluation classiques de précision (P), rappel (R) et f-mesure (F), qui sont génériques et faciles à interpréter (Martin et al., 2004). Le tableau 2 montre les résultats de l'évaluation de notre approche par rapport à celle de l'état de l'art. On constate sans surprise que notre approche est meilleure que celle de l'état de l'art (F=87% vs. F=68%). Cela est dû au fait que nous avons : (i) gardé la structure telle qu'elle de la question posée par l'utilisateur, et (ii)

Mesures	Notre approche	Bouziane et al. (2018)
P	**0.88**	0.71
R	**0.86**	0.66
F	**0.87**	0.68

TAB. 2 – *Résultats d'évaluation*

analysé les relations de dépendances pour comprendre l'intention de l'utilisateur. Ces informations linguistiques sont ensuite gérées par des règles formalisées.
Cependant, la méthode de (Bouziane et al., 2018) est fondée sur la modification de la structure de la question (suppression des mots vides), ce qui fournit une réponse incorrecte. La suppression des mots vides détruit le sens de la phrase ; ex. "Je mange une pomme de terre" vs. "mange", "pomme", "terre".
Notre méthode améliore d'une manière significative les systèmes QR de l'état de l'art.

5 Conclusion et perspectives

L'approche proposée dans cet article s'appuie sur la combinaison de méthodes de TALN et des technologies du Web Sémantique pour fournir des réponses aux questions exprimées en langage naturel. L'idée consiste à d'abord étiqueter les questions par des informations représentant la structure syntaxique (verbes, conjonctions, adjectif, etc.). Par ailleurs, notre travail permet de définir quatre règles exploitant les relations syntaxiques dans une question. Ces règles sont ensuite traduites en requêtes SPARQL exprimées suivant le vocabulaire d'une ontologie. Nous avons ici mis l'accent sur la qualité des réponses générées. Ce travail a été évalué sur un jeu de données, montrant des résultats prometteurs.
Nous proposons d'améliorer notre méthode comme suit : (i) retravailler l'intention du robot pour qu'il traite à plusieurs intentions à la fois, (ii) formaliser d'une manière (semi)-automatique les règles définies, (iii) introduire la noition d'alignement des ontologies afin de traiter différents domaines et d'améliorer les résultats, et (iv) ajouter l'aspect vocal.

Références

Albarghothi, A., F. Khater, et K. Shaalan (2017). Arabic question answering using ontology. *Procedia Computer Science 117*, 183–191.

Arain, A., A. Manzoor, K. Brohi, K. Haseeb, I. Halepoto, et I. Korejo (2018). Artificial intelligence mark-up language based written and spoken academic chatbots using natural language processing. *Sindh University Research Journal-Science Series 50*, 153–158.

Auer, S., C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, et Z. Ives (2007). Dbpedia: A nucleus for a web of open data. Volume 6, pp. 722–735.

Bansal, R. et S. Chawla (2014). An approach for semantic information retrieval from ontology in computer science domain. *International Journal of Engineering and Advanced Technology 4*, 58–65.

Barskar, R., G. Ahmed, et N. Barskar (2012). An approach for extracting exact answers to question answering (qa) system for english sentences. *Procedia Engineering 30*, 1187–1194.

Bouziane, A., D. Bouchiha, N. Doumi, et M. Malki (2018). Toward an arabic question answering system over linked data. *Jordanian Journal of Computers and Information Technology*.

Chiticariu, L., Y. Li, et F. R. Reiss (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 827–832.

Ferré, S. (2016). Sparklis: An expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web 8*, 405–418.

Kulkarni, C. S., A. U. Bhavsar, S. R. Pingale, et S. S. Kumbhar (2017). Bank chat bot - an intelligent assistant system using nlp and machine learning. *International Research Journal of Engieering and Technology 4*, 2374–2376.

Martin, A., J. Garofolo, J. Fiscus, A. Le, D. Pallett, M. Przybocki, et G. Sanders (2004). Nist language technology evaluation cookbook.

Miller, G. A. (1995). Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM 38*, 39–41.

Ngonga Ngomo, A.-C., L. Bühmann, C. Unger, J. Lehmann, et D. Gerber (2013). Sorry, i don't speak sparql: Translating sparql queries into natural language. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pp. 977–988. ACM.

Pradel, C., O. Haemmerlé, et N. Hernandez (2014). Swip: A natural language to sparql interface implemented with sparql. In *Graph-Based Representation and Reasoning*, pp. 260–274.

Rajosoa, M., R. Hantach, S. B. Abbes, et P. Calvez (2019). Hybrid question answering system based on natural language processing and sparql query. *The 3rd International Workshop on the Applications of Knowledge Representation and Semantic Technologies in Robotics (AnSWeR19) 2487*, 94–102.

Yao, X. et B. V. Durme (2014). Information extraction over structured data: Question answering with freebase. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, Volume 1, pp. 956–966.

Summary

Chatbot is a conversational agent that communicates with users based on natural language. It is founded on a question answering system, dealing with user's intent. Several recent works have been proposed with some limitations. The originality of our approach is to combine the natural language processing methods and semantic web techniques. A domain ontology used as a knowledge base for describing informations in RDF triplestore. First experimental results prove the interest of our proposition.