Moteur de questions-réponses d'une base de connaissances

Michel Plu*, Johannes Heinecke*

*Orange Labs
2 avenue Pierre Marzin
22307 Lannion cedex
{michel.plu, johannes.heinecke}@orange-ftgroup.com

Résumé. Cet article présente comment la gestion et l'exploitation de connaissances issues du site web Wikipedia ont permis de développer une telle fonction qui a été intégrée depuis février 2010 dans un moteur de recherche internet français pour le grand public. Aujourd'hui cette fonction est capable de répondre à des questions formulées en langage naturelle sur environs 170 000 lieux ou personnes. La formalisation des données extraites de wikipedia en connaissances au format OWL ou RDFS a permis de déduire de nouvelles informations manquantes, de typer les entités nommées trouvées et de traiter de nouvelles formes de questions qui étaient non traitées.

1 Introduction

Une nouvelle tendance des moteurs de recherche sur le web est d'enrichir leur liste réponses en répondant directementă aux questions posées dans les requêtes des utilisateurs. Par exemple à une requête comme « population de la région bretagne », un tel moteur de recherche affiche en première réponse « 3 120 288 habitants » alors qu'un moteur de recherche plus classique ne fournirait qu'une liste de documents.

L'utilité d'une telle fonction appelée par la suite moteur de questions réponses est évidente. En trouvant directement la réponse à sa question, l'utilisateur gagne du temps et est encouragé à poser d'autres questions. En effet, sans cette fonction l'utilisateur est censé parcourir les documents ou leurs extraits proposés dans la liste réponses pour éventuellement trouver la réponse attendue. Ce parcours de documents est le plus souvent fastidieux et inconfortable pour l'utilisateur qui doit chercher lui même la réponse. Cette pénibilité est encore plus grande lorsque le terminal est un téléphone mobile.

Le développement d'un moteur de questions réponses comporte de nombreuses difficultés. Lorsque celui-ci utilise une base de données pour répondre aux questions, il faut tout d'abord transformer une requête constituée de mots clés ou parfois d'une phrase en langage naturel en une requête formelle compréhensible par le système de gestion de base de données utilisé. Ensuite, pour répondre le plus précisément possible à un maximum de réponses, il faut d'une part disposer d'un large ensemble de connaissances et d'autre part être capable de reconnaître le plus possible de formes de requêtes correspondant aux réponses que l'on peut produire.

Cet article présente le développement d'un tel moteur de questions réponses pour un moteur de recherche web grand public accessible depuis trois portails internet français. La section

suivante commence par décrire l'architecture du moteur. La section 3 présente l'utilisation de mécanismes d'inférences pour qualifier et étendre l'ensemble des connaissances disponibles. Enfin la section 4 conclu cet article et liste quelques perspectives d'améliorations dans le cadre d'une vision plus générale d'une interface utilisateur sur le web sémantique.

2 Architecture du moteur questions-réponses

2.1 Le moteur de questions réponses (front-office)

L'architecture globale du moteur de question réponse qui est interrogé est présentée dans la figure 1. L'utilisateur saisit d'abord sa requête textuelle via un navigateur web. Ce texte

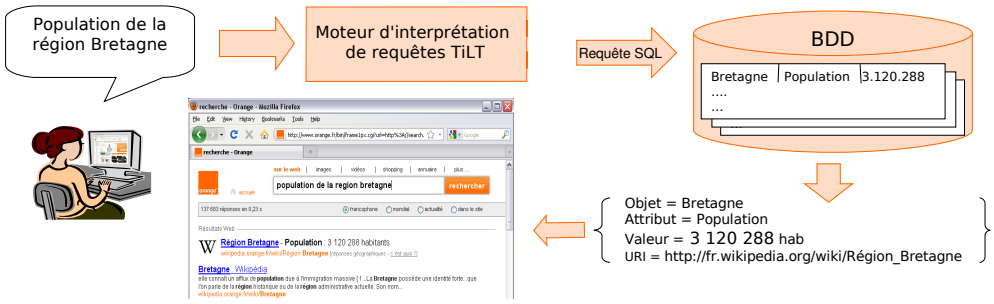

FIG. 1 – *Architecture du moteur questions-réponses*

est ensuite analysé par un interpréteur de requêtes. Cet interpréteur est développé à partir de la plateforme de traitement automatique du langage naturelle Tilt (Heinecke et al., 2008). Il permet la transformation d'une requêtes à base de mots en une requête formelle soumise au soumise à une base de données. La réponse est alors transformée pour être insérée au debut de la liste réponse du moteur affichée dans le navigateur internet.

2.2 La génération des données (back-office)

Les données exploitées par le moteur de questions réponses sont générées à partir d'un système de gestion de base de connaissances (cf. fig. 2). Ces données sont d'une part celles stockées par la base de données et d'autre part des données linguistiques utilisées par l'interpréteur de requêtes. La base de connaissances utilisée est gérée par une base de données Oracle étendu par un module sémantique (Oracle, 2010). Les connaissances sont modélisées sous la forme de triplets RDF (*http://www.w3.org/RDF/*). Un triplet RDF est constitué d'un sujet, d'une propriété et d'une valeur aussi appelée objet. Chacun de ces éléments est défini par une URI (*http://fr.wikipedia.org/wiki/Uniform_Resource_Identifier*). L'objet d'un triplet peut aussi être une chaine de caractères quelconque. La base de connaissances utilisée gère actuellement plus de 13 millions de triplets. Ces triplets proviennent d'une part du projet DBpedia (*http://dbpedia.org/About*) à partir duquel nous avons récupéré : une ontologie en OWL (*http://www.w3.org/TR/owl-features/*), la classification des URI utilsés et un ensemble de valeurs de propriétés de certains sujets issus d'une analyse automatique des infobox des pages

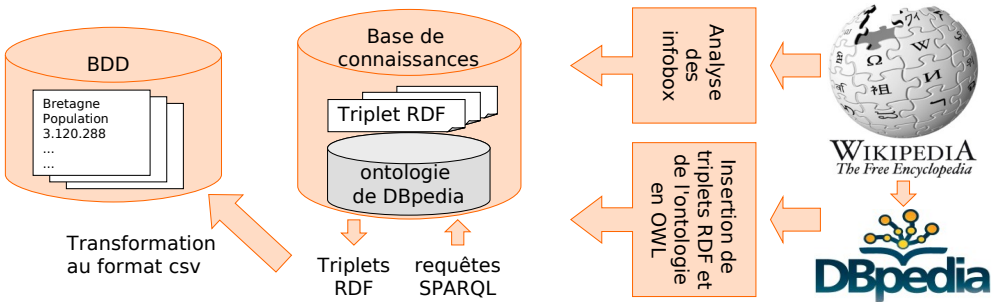

FIG. 2 – *Architecture de la base de connaissances*

du site français de Wikipedia (*http://fr.wikipedia.org*). L'infobox d'une page Wikipedia correspond à un tableau de la page ayant un modèle (template) bien structuré et synthétisant des propriétés caractéristiques du sujet traité dans la page.

Ces triplets décrivent environs 3,4 millions d'instances et ont étés complétés a partir d'une autre analyse automatique que nous avons développée pour les infobox de pages de Wikipedia correspondant à des lieux et à des personnes pour lesquelles nous voulions avoir plus d'informations. Ceci a permis d'ajouter des nouvelles informations sur plus de 26 000 lieux et plus de 32 000 personnes.

La base de données interrogée par le moteur de question réponse peut être régulièrement mise à jour à partir de requêtes sur cette base de connaissance. Le support par celle-ci de requêtes en SPARQL (*http://www.w3.org/TR/rdf-sparql-query/*) offre à la fois une grande flexibilité et la capacité d'étendre sémantiquement la sélection de données. Par exemple une requête de l'ensemble des propriétés de tous les sujets de type « Person » permet de récupérer des sujets de types plus spécifiques tels que « Actor » et « Musician » et l'ensemble de leurs propriétés disponibles dans la base même si celles-ci sont nouvelles et définies pour très peu de sujets.

3 Gestion de la base de connaissances

3.1 Le besoin d'inférences

Nous avons vu précédemment que les connaissances interrogées par le moteur de question réponses sont issues du site web Wikipedia et de connaissances produites par le projet DBpedia aussi issues du même site. Le principal intérêt des connaissances réutilisées de ce projet est la classification des sujets traités dans des classes organisées au sein d'une ontologie écrite dans un format standard (OWL). Malheureusement cette classification n'est faite que pour une partie des sujets traités dans les pages francophones de Wikipedia. Dans notre collecte seulement 66% des sujets avaient un type défini dans les connaissances de DBpedia.

De plus, certaines entités peuvent être aussi trouvées comme valeur d'une propriété sous la forme d'une chaîne de caractères et non d'une URI. Or cette entité ne correspond pas toujours à un sujet d'une page existante de Wikipedia. Elle ne peut donc pas avoir d'identifiant (URI) dans DBpedia et il peut donc être utile d'en créer une pour notre usage dans le moteur de questions réponses. Ces nouveaux sujets doivent aussi être classés dans l'ontologie de DBpedia.

De même très peu des propriétés issues des infobox des pages françaises de Wikipedia sont définies dans l'ontologie. Dans les données réutilisées de DBpedia seulement 13,5% des propriétés utilisées pour décrire les sujets avaient des propriétés « domain » et « range » définies. Pour compléter ces connaissances manquantes nous avons définis un mécanisme d'induction permettant de déduire les valeurs des propriétés « domain » et « range » de certaines propriétés à partir du type connu de sujets et d'objets de triplets présents dans la base. Rappelons que ces propriétés appelées par la suite respectivement domain et range, définissent repsectivement la classe des sujets et des objets d'une propriété. Nous les utilisons pour inférer le la classe de certaines URI comme nous le verrons ci-dessous.

Afin de pouvoir répondre à un maximum de question il est aussi intéressant de pouvoir connaître les valeurs d'un maximum de propriétés. Or certaines valeurs de propriétés peuvent être déduites à partir de connaissances existantes. Ces déductions peuvent être réalisées à partir de l'écriture spécifique de règles d'inférences en chainage avant ou en exploitant la sémantique des propriétés définies dans RDFS ou OWL[1].

3.2 Classification d'entités

La classification d'entités gérées par la base de connaissances se fait à partir de la valeur des propriétés « domain » et « range » des propriétés. En effet la sémantique de ces propriétés implique respectivement que les sujets et respectivement les objets de ces propriétés sont de la classe définie par la valeur de ces propriétés « domain » et respectivement « range ».

Pour pouvoir classer ainsi le maximum d'entités nous avons cherché à calculer automatiquement les « domain » et « range » de propriétés qui ne sont pas dans les connaissances initiales. Afin de connaître le domaine (resp. range) d'une propriété, nous allons utiliser les triplets avec cette propriété pour lesquelles la classe du sujet (respectivement de l'objet) est connue. Nous allons ensuite regrouper pour chaque propriété les classes des sujets (resp. object) trouvés et conserver le nombre de triplets correspondant.

A l'aide de la hiérarchie de classes de l'ontologie de Dbpedia, les classes des sujets (resp des objects) d'une propriété sont remplacés par leur ancêtre commun le plus spécifique. Prenons par exemple, la propriété « paysDeRésidence » dont voici les classes de sujets trouvés : *BritishRoyalty*, *Writer,Artist*, *Person*, *Scientist*, *OfficeHolder*, *Actor*, *PrimeMinister*, *MemberOfParliament*, *Journalist*, *Criminal*. Toutes ces classes sont des sous-classes de la classe *Person* elles seront donc remplacées par la classe *Person*. Le domain proposé pour cette propriété sera donc la classe Person et sera justifiée par la somme des triplets associés à chacune des sous classes trouvées. Pour filtrer certaines propositions liées a des erreurs dans les infobox de wikipedia, seules celles ayant une justification au dessus d'un certain seuil sont conservées.

Dans notre base de connaissance collectée on a ainsi réussit à calculer le domain de 5 691 propriétés et le range de 850. Avec un filtrage du nombre de triplets supérieur à 10 et devant représenté plus de 60% des triplets, on a réussi à valider automatiquement 1 959 valeur de domain et 81 valeurs de range

Une fois le domain et le range connu il est alors possible de classer les entités trouvés comme sujet ou objet d'un triplet. Mais plusieurs classes peuvent être parfois possible pour une même URI. On ne conserve alors que l'ancêtre commun le plus spécifique de ces classes. Ce mécanisme d'inférence nous a permis de classifer 4 134 entités de notre base.

[1] Cf. *http://www.w3.org/TR/rdf-mt/#RDFSExtRules*

La connaissance du range de propriétés permet aussi de transformer les objets de triplets de propriété qui au lieu d'être des identifiant d'entités (des URI) sont des chaînes de caractères quelconques. Pour cela, pour chacune de ces propriétés, on recherche une instance dont la classe est une sous classe du range de la propriété et dont la valeur de l'une de ses propriétés label est similaire à la chaîne de caractères trouvée comme objet de la propriété. Plusieurs mesures de similarité entre chaine sont utiliséesă : similarité de préfixe, distance d'edition, similarité phonétique (Euzenat et Shvaiko, 2007). Lorqu'une telle substitution peut se faire la chaine de caractère initiale est ajoutée comme objet d'une nouvelle propriété label pour l'entité trouvée afin d'avoir plus d'exemples de nommage de celle-ci. Cette transformation va permettre de faire des jointures entre triplets nécessaires notamment nécessaire pour faire certaines inférences à partir de règles de déduction ou répondre à des requêtes plus complexes de la forme « quel est le lieu de naissance de la femme de Brad Pitt ? ».

Ces mécanismes a permis de transformer 2 601 valeurs de triplet en une URI connue de la base.

3.3 Inférences de nouvelles connaissances

Toujours dans l'objectif de pouvoir répondre à plus de questions, nous avons identifié le besoin de pouvoir compléter des valeurs absentes de propriétés pour certains sujets ou de définir de nouvelles propriétés correspondant à des questions des utilisateurs. Afin de faciliter l'ajout de tels calculs par des personnes responsables de l'édition du service de questions réponses, il est nécessaire de séparer leur définition du code du logiciel de gestion de la base de connaissances. Ces calculs sont donc définis de manière déclarative par la définition de règles de productions ou par la définition en RDFS ou OWL de propriétés sur les propriétés à calculer. Afin de s'affranchir des contraintes syntaxiques de chaque format, une interface utilisateur permet d'éditer ces données. L'exemple de règles suivant illustre le type d'inférence réalisée :

```
[ r_entraineur: (?joueur <http://dbpedia.org/property/club> ?club)
                (?club <http://dbpedia.org/property/club> ?entraineur)
  → (?joueur <http://dbpedia.org/property/clubEntraineur> ?entraineur)]
```

Cette règle permet de compléter la propriété désignant l'entraîneur d'un joueur à partir de la valeur de la propriété désignant l'entraîneur de son club. Ces règles sont exécutées par le moteur de règles en chainage avant intégré au framework Jena (*http://jena.sourceforge.net/inference/#RULEforward*).

Une autre manière de faire de l'inférence est de définir des relations de symétrie et d'inverse de propriétés. Ceci peut s'exprimer avec le langage OWL Lite (*http://www.w3.org/TR/2004/REC-owl-features-20040210/#s2.1*). Par exemple, la définition que la relation conjoint est symétrique permet de compléter la valeur non définie de cette propriété pour certains sujets. Cela a ainsi permis de déduire la valeur manquante de plus de 3 500 propriétés *conjoint*. Autre exemple, le fait les propriétés *film_notable* et *acteur* sont des relations inverses ont permis de compléter la liste des films de certains acteurs. Ces propriétés en OWL sont exploitées par le moteur de raisonnement OWL du framework Jena.

4 Conclusion et perspectives

Nous avons présenté dans cet article comment exploiter les capacités d'une base de connaissances pour étendre les capacités d'un moteur de questions réponses. Les connaissances formelles permettent la production de nouvelles données par différents mécanismes d'inférences présentés. La génération automatique de données linguistiques pour l'interpréteur de requête et des données interrogées par le moteur de question réponses utilisent le langage de requête sémantique SPARQL. Ceci offre à la fois une grande flexibilité et la capacité d'étendre sémantiquement la sélection de données. Ces mécanismes ont été validés dans un cas réel avec une base de connaissance de plus de 13 milions de triplets. Une particularité notoire de ces travaux est leur mise en opération dans le moteur de recherche de trois portails internet français (*http://www.orange.fr*, *http://voila.fr*, *http://lemoteur.fr*) avec une capacité de traitement de plus de 300 requêtes à la seconde. Comparativement aux moteurs de recherche du web les plus couramment utilisés, ce moteur de question réponses est capable de répondre à beacoup plus de questions grace aux connaissances inférées et à l'interprétation linguistique des requêtes qui supporte de multiples formulations possibles d'une même question y compris celles comprenant certaines erreurs.

Néanmoins, nous considérons cette réalisation comme un embryon de notre vision de ce que devrait être un futur moteur de recherche pour le web sémantique en plein développement.

Remerciements

Nous remercions nos collègues qui ont contribué aux travaux décrit dans cette article : Emilie Guimier De Neef, Fanny Parganin, Arnaud Debeurme et les équipes de développement du moteur de recherche *http://orange.fr*.

Références

Euzenat, J. et P. Shvaiko (2007). *Ontology Matching*. Heidelberg : Springer.

Heinecke, J., G. Smits, C. Chardenon, E. Guimier De Neef, E. Maillebuau, et M. Boualem (2008). TiLT : plate-forme pour le traitement automatique des langues naturelles. *TAL 49 :2*, 17–41.

Oracle (September 2010). Oracle database semantic technologies overview. *http://download.oracle.com/docs/cd/E11882_01/appdev.112/e11828/sdo_rdf_concepts.htm#CIHHEDAC*.

Summary

This article presents how the knowledge extracted from Wikipedia is used to develop a question-answering system, integrated in a search engine since February 2010. Currently the engine responds on questions on about 170,000 localities and persons. The provided answers are extracted from a knowledge base is formalized using the OWL and RDFS standards.