Une approche de recherche d'attributs pertinents pour l'agrégation d'information

Ines Krichen, Arlind Kopliku, Karen Pinel-Sauvagnat et Mohand Boughanem

Université de Toulouse, IRIT UMR 5505 CNRS
118 route de Narbonne, F-31062 Toulouse cedex 9

RÉSUMÉ. *Nous présentons dans cet article une approche d'agrégation de résultats de recherche basée sur la détection d'attributs pertinents répondant à une requête de type classe ou entité nommée. L'agrégation permet de renvoyer à l'utilisateur un ensemble d'informations bien organisées, générées à partir de plusieurs documents, au lieu d'une liste de documents qui répondent chacun à une partie de son besoin. Notre approche s'appuie sur 3 étapes qui sont la sélection des entités et attributs pertinents à la classe, leur filtrage et le tri des attributs pertinents. A l'issue de ces étapes, les résultats sont visualisés dans un tableau. Afin de tester cette approche, nous avons mis au point une application dans le but d'évaluer la qualité des tableaux retournés à partir de la base de connaissance DBpedia. L'évaluation montre que notre approche offre aux utilisateurs jusqu'à 6 attributs pertinents parmi 10 dans les tableaux retournés.*

ABSTRACT. *We present in this paper a results aggregation approach based on relevant attributes detection and answering to class or entity-queries. Instead of returning the traditional list of documents, aggregation allows to return to users well-organized information, coming from different sources. Our approach is based on three steps: entities and attributes selection, filtering and attributes ranking. Results are then organized in a table form. In order to evaluate our approach we developed an application that aims to assess the quality of the returned tables constructed thanks to DBpedia. The evaluation shows that our approach is able to retrieve up to 6 out of 10 relevant attributes in the returned tables.*

MOTS-CLÉS : *Recherche d'information, Agrégation de résultats , Recherche d'attributs*

KEYWORDS: *Information Retrieval, Aggregated search, Attribute search*

1. Introduction

Les Systèmes de Recherche d'Information (SRI) renvoient en réponse à une requête une liste de documents potentiellement pertinents. L'information pertinente recherchée peut se retrouver entièrement dans un document ou être éparpillée dans plusieurs documents (Boughanem *et al.*, 2008). L'utilisateur doit alors parcourir la liste des documents sélectionnés et regrouper et sélectionner les parties ou fragments d'information qu'il juge pertinents. Un fragment d'information peut être une définition, une image, un texte descriptif, une vidéo, un tableau, ou même un attribut avec sa valeur (par exemple une adresse, un téléphone, etc.). La réponse pertinente idéale serait donc composée de tous les fragments sélectionnés par l'utilisateur.

De nombreux besoins d'information nécessitent l'agrégation d'information à partir de plusieurs sources (documents). Par exemple, pour les requêtes de type entités nommées (*"Toulouse"*) ou classe d'entités (*"villes de France"*), qui représentent 71% des requêtes posées sur le Web d'après (Kato *et al.*, 2009), la liste n'est vraiment ni la meilleure ni la plus naturelle des réponses. En effet, pour la requête *"restaurant chinois à New York"*, retourner tous les documents qui contiennent les termes *"restaurant" "chinois"* et *"New York"* risque de faire passer la faim à l'utilisateur. Pour répondre à cette requête une alternative est d'extraire et regrouper les données nécessaires (voir figure1).

Figure 1. *Exemple de réponse agrégée*

Nous proposons dans cet article de traiter la problématique de l'agrégation d'information, en utilisant une approche alternative où l'information est organisée autour de l'entité et non du document. Notre approche a pour but d'extraire et trier les informations représentatives d'une entité ou d'une classe donnée à partir de plusieurs documents. Plus particulièrement, nous nous intéressons dans cette étude aux requêtes de type classe d'entités. Pour représenter les informations nous adoptons une des formes

les plus répandues dans la recherche agrégée, celle des tableaux. Ces tableaux résultats contiennent un ensemble d'entités avec leurs attributs. Afin de construire ces tableaux résultats, les problématiques à traiter sont les suivantes :

– sélectionner et trier les entités représentatives de la classe,

– sélectionner et trier les attributs descriptifs de la classe,

– trouver les valeurs de ces attributs pour chacune des entités sélectionnées.

Dans cet article, nous nous intéressons principalement à la deuxième problématique. Afin de tester notre approche, nous utilisons comme source de données Wikipedia, et plus particulièrement DBpedia.
Cet article est organisé comme suit. La section 2 est consacrée à la présentation de l'état de l'art sur la recherche agrégée. La section 3 présente notre approche pour l'agrégation d'information. La section 4 concerne l'évaluation et nous concluons et énonçons quelques perspectives en section 5.

2. Etat de l'art : Recherche agrégée

La recherche agrégée a été définie pour la première fois dans un atelier à SIGIR 2008 (Murdock *et al.*, 2008), comme étant une tâche cherchant à rassembler des informations provenant de sources différentes, et à les présenter dans une seule interface. Des informations de différents types (image, vidéo, etc...) et de différentes granularités (entités, attributs, etc...) sont reliées et parfois même combinées par une ou plusieurs relations logiques (tableau, bloc, etc...) afin de composer un résultat agrégé.
La recherche agrégée est un domaine de recherche récent qui se base sur des approches de recherche d'information existantes. Parmi les problématiques qui doivent être résolues, nous pouvons citer :

– la redondance d'information qui est une conséquence naturelle du traitement des grandes collections,

– le choix du degré de granularité attendu par l'utilisateur (paragraphe, document, entité, attribut, ...),

– l'organisation des résultats de recherche dans l'espace de visualisation,

– la mise en évidence des différentes relations entre les contenus.

Il existe différentes techniques pour l'agrégation que nous pouvons classer en deux grandes approches : la fusion de différentes recherches verticales et la construction d'un document à partir de plusieurs documents.
Dans les approches du premier groupe, les différentes recherches verticales renvoient des informations d'un seul type de support (image, vidéo, etc ...) ou de contenu (news, livre, etc ...). Ces informations sont ensuite triées et agrégées pour être présentées à l'utilisateur. C'est par exemple ce que font les moteurs de recherche commerciaux Google Universal[1], Yahoo Alpha[2], ou encore Bing[3].

1. http ://www.google.fr/
2. http ://au.alpha.yahoo.com/
3. http ://www.bing.com/

Les approches du second groupe cherchent à construire ou à générer automatiquement un document à partir de plusieurs documents de même source ou de sources différentes. Pour ce faire, plusieurs méthodes ont été proposées dans la littérature, parmi lesquelles nous pouvons citer (Paris *et al.*, 2010) (Sauper *et al.*, 2009) (Cafarella *et al.*, 2008) ou (Elmeleegy *et al.*, 2009). Les auteurs de (Sauper *et al.*, 2009) cherchent à construire automatiquement des articles médicaux pour Wikipedia à partir de modèles qu'ils ont eux-mêmes générés. Ils ont montré que l'intégration d'informations structurées provenant d'articles déjà existants dans Wikipedia dans le processus d'apprentissage d'extracteurs de contenus améliore la qualité des articles construits.
D'autre travaux ont exploité les structures contenues dans le Web comme les tableaux et les listes. Par exemple, les auteurs de (Cafarella *et al.*, 2008) ont élaboré un projet, intitulé WebTables, dont le but est de détecter, classifier et filtrer les tables relationnelles du Web afin d'agréger les données retournées dans un document final. L'inconvénient de cette approche est que la détection des tables relationnelles dans le Web n'est pas toujours évidente.
De son côté, GoogleLabs[4] a lancé un outil expérimental, intitulé Google Squared, qui permet de générer un tableau descriptif pour une requête donnée (voir exemple sur la figure 2).

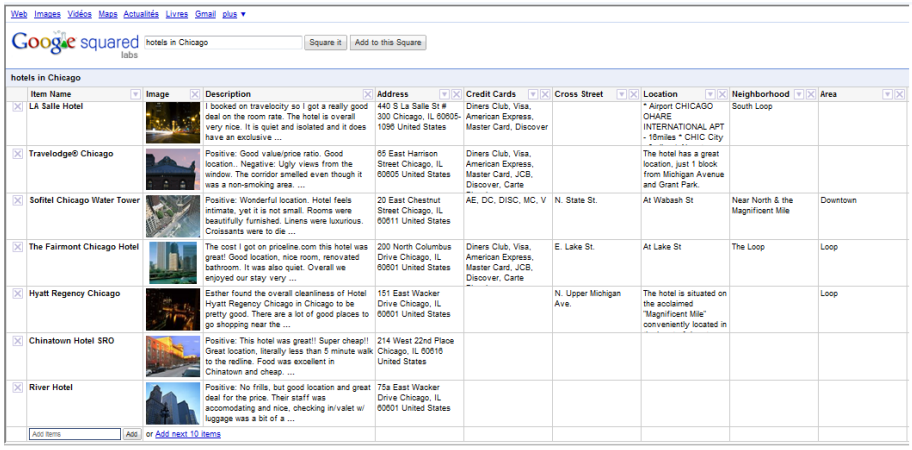

Figure 2. *Résultats de la requête "hotels in Chicago" pour le moteur de recherche Google Squared*

Malgré son originalité, cette dernière approche retourne souvent des tableaux troués et renvoie parfois des attributs avec des valeurs erronées.

4. http ://www.googlelabs.com

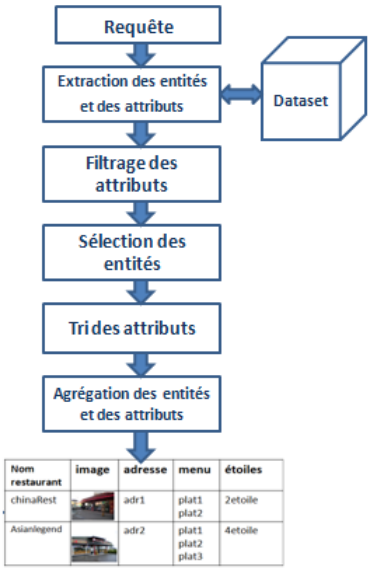

Figure 3. *Approche générale*

Un autre système, ListExtract, a été proposé par (Elmeleegy *et al.*, 2009). C'est une technique d'extraction des tables à partir des listes. La technique est indépendante du domaine étudié sauf qu'elle ne traite que les listes.

3. Une approche pour l'agrégation d'information

L'approche d'agrégation que nous proposons s'intéresse à des requêtes de type entité nommée ou classe d'entités. Son objectif est d'élaborer un tableau qui contient une ou plusieurs entités avec leurs attributs et leurs valeurs associées. Dans ce travail, nous nous limitons aux requêtes de type classe en considérant que nous pouvons ramener les entités à la forme d'une classe en déterminant celle qui leur ressemble le plus.
Notre approche comporte plusieurs étapes, schématisées sur la figure 3. La requête est tout d'abord soumise à une base documentaire. La première étape consiste à extraire les entités et les attributs correspondant à la requête. La quantité importante d'informations obtenues nécessite une étape de filtrage des attributs, ainsi qu'une étape de sélection des entités importantes qui participeront ensuite au tri des attributs. Finalement nous agrégeons les entités et les attributs obtenus pour construire notre résultat final, sous forme de tableau.
Nous détaillons toutes ces étapes dans ce qui suit.

3.1. *Extraction des entités et des attributs*

Cette première étape consiste à collecter les entités et les attributs potentiellement pertinents pour une requête de type classe. Ces informations sont, dans notre étude, collectées à partir de Wikipedia et plus particulièrement de DBpedia[5]. Wikipédia, encyclopédie libre, collaborative et multilingue[6], contient de l'information compréhensible, de haute qualité et bien structurée. Parmi ces informations structurées, on peut citer les catégories et les listes, l'accès à l'historique et les infobox[7] (Medelyan *et al.*, 2009).
Nous avons choisi d'aborder les données de Wikipedia au travers de DBpedia. DBpedia convertit le contenu de Wikipedia en des données structurées en utilisant les techniques du Web sémantique (Auer *et al.*, 2007). Cette gigantesque base de connaissances couvre de nombreux domaines spécifiques (Bizer *et al.*, 2009). A partir de DBpedia, nous récupérons l'ensemble des entités qui appartiennent à la requête c'est-à-dire à la classe c en question. Pour ce faire, nous avons utilisé un des moyens d'accès à DBpedia, SPARQL Endpoint, qui consiste à envoyer une requête SPARQL au service offert par DBpedia (http ://dbpedia.org/sparql), qui renvoie ensuite un document comportant des liens vers les entités de la classe c. On trouvera un exemple de requête sur la figure 4.

```
SELECT distinct ?entities ?label
        WHERE { {?entities rdf:type <http://dbpedia.org/class/yago/AdventureNovels>}
            UNION
            {?entities rdf:type <http://dbpedia.org/ontology/AdventureNovels>}
            UNION
            {?entities skos:subject <http://dbpedia.org/resource/Category:AdventureNovels>}
            ?entities rdfs:label ?label.
             FILTER ( lang(?label) = 'en' ) }
```

Figure 4. *Exemple de requête SPARQL récupérant les entités correspondant à la classe 'AdventureNovels'*

Pour augmenter le nombre d'entités renvoyées pour chaque requête c, nous interrogeons les 3 types de catégorisation offerts sous DBpedia qui sont :

– YAGO (Suchanek *et al.*, 2007), qui contient 286000 classes qui forment une arborescence hiérarchique. C'est une ontologie généraliste de grande taille et extensible qui s'appuie sur les entités et attributs extraits à partir de Wikipedia.

– Skos[8], qui contient 415000 classes qui sont les catégories offertes par le système de Wikipedia. Le problème avec cette catégorisation est qu'elle représente une faible relation de parenté entre les articles (les entités).

5. http ://wiki.dbpedia.org
6. http ://en.wikipedia.org/wiki/History_of_Wikipedia
7. Modèle qui permet d'insérer du texte ou du code wiki dans une page, de manière automatisée, dynamique et configurable
8. http ://www.w3.org/2004/02/skos/

– OWL-DBpedia[9], qui contient 205 classes, sous forme d'une hiérarchie peu profonde, et 1210 propriétés construites manuellement à partir de 350 infobox les plus utilisés au niveau de la version anglaise de Wikipedia.

Les entités résultats (voir figure 5 pour un exemple), ainsi renvoyées pour la classe, sont ensuite utilisées pour extraire les attributs de la classe et leurs valeurs. Pour cela, nous utilisons un autre moyen d'accès à DBpedia, "Data Link", qui consiste à accéder directement au lien de l'entité concernée. Ce lien constitue un identifiant de l'entité au sein de la source DBpedia. Par exemple "http ://dbpedia.org/page/Treasure_Island" est un identifiant unique pour l'entité *"Treasure_Island"*.

SPARQL results:

entities	label
:Adventures_of_Huckleberry_Finn	"Adventures of Huckleberry Finn"@en
:Moonfleet	"Moonfleet"@en
:The_Farther_Adventures_of_Robinson_Crusoe	"The Farther Adventures of Robinson Crusoe"@en
:Treasure_Island	"Treasure Island"@en
:The_Toll-Gate	"The Toll-Gate"@en
:Rupert_of_Hentzau	"Rupert of Hentzau"@en
:Peter_Duck	"Peter Duck"@en
:Coot_Club	"Coot Club"@en
:Missee_Lee	"Missee Lee"@en
:Swallowdale	"Swallowdale"@en
:White_Fang	"White Fang"@en
:Sard_Harker	"Sard Harker"@en
:Beau_Geste	"Beau Geste"@en
:The_Coral_Island	"The Coral Island"@en
:Gentlemen_of_the_Road	"Gentlemen of the Road"@en
:The_Maracot_Deep	"The Maracot Deep"@en
:Journey_to_the_River_Sea	"Journey to the River Sea"@en
:The_Big_Six	"The Big Six"@en
:Pigeon_Post	"Pigeon Post"@en
:ODTAA	"ODTAA"@en
:The_Riddle_of_the_Sands	"The Riddle of the Sands"@en

Figure 5. *Exemple d'entités résultats pour la classe 'AdventureNovels'*

Chaque entité de DBpedia, reliée à article de Wikipedia, est décrite par un ensemble de propriétés générales et un ensemble de propriétés infobox-spécifiques (Bizer *et al.*, 2009). Ces propriétés sont utilisées pour extraire les attributs des entités et leur valeur. Les propriétés spécifiques sont spécifiques à l'entité en question, trouvées généralement dans l'infobox. Les propriétés générales sont celles qui se répètent dans toutes les entités quelque soit leur nature. Elles incluent une étiquette, un court et un long résumé en anglais, un lien à l'article correspondant de Wikipedia, des geo-coordonnées (si disponibles), un lien à une image de l'entité, des liens aux pages Web externes, des liens internes aux entités de DBpedia, ect.... Si une entité est décrite en plusieurs langues, alors il aura des résumés courts et longs dans ces langues.

9. http ://wiki.dbpedia.org/Ontology

3.2. *Filtrage des attributs*

Les propriétés génériques et spécifiques, décrites dans la section précédente, contiennent un nombre important d'informations utiles et informatives sur les entités en question. Elles comportent cependant également de l'information répétitive, inutile, de mise en forme et de catégorisation que l'utilisateur n'a pas besoin de consulter. En effet, nous pouvons trouver la même valeur reliée à plusieurs attributs soit de noms différents (*birthdate* et *birthDate*), soit de niveaux différents (http ://dbpedia.org/ontology/Place/location et http ://dbpedia.org/ontology/location), soit carrément d'origines différentes (http ://dbpedia.org/ontology/location et http ://dbpedia.org/property/location).
Une étape de filtrage est donc nécessaire. Elle consiste tout d'abord à ne garder que les attributs en anglais (car nous ne produisons que des documents en anglais). Ensuite, nous éliminons les attributs et les valeurs redondants, les attributs de mise en forme (*wikiPageUsesTemplate*) et de catégorisation (*rdf :type, skos :subject*) ainsi que les attributs vides[10] comme *redirect, display, disambiguates, same as*. Nous éliminons aussi les propriétés venues de ressources externes comme le dataset geo-name[11] et GeoRSS[12]. L'étape de filtrage permet de minimiser le nombre d'informations non pertinentes et d'augmenter les chances des propriétés spécifiques, importantes et avec un caractère dominant à être classées devant les propriétés générales. Le classement des attributs est détaillé dans la section 3.4.
A l'issue de cette étape de filtrage, nous définissons pour chaque entité de la requête classe c un ensemble d'attributs avec leurs valeurs.

3.3. *Sélection des entités*

Notre but, dans ce travail, n'est pas de classer les entités mais plutôt de classer les attributs. Cependant, le tri des attributs est très relié à celui des entités : nous supposons qu'un bon attribut est généralement présent dans la plupart des entités importantes. Afin d'analyser l'impact des entités sur le classement des attributs, nous avons mis en place trois algorithmes de sélection d'entités.

Le premier, intitulé *Shorty*, sélectionne les n premières entités qui ont le moins d'attributs parmi celles qui ont au moins un attribut (voir algorithme 1). L'intérêt de cette sélection est de favoriser les attributs des entités qui ont peu d'attributs. L'hypothèse derrière ce choix est que le peu d'attributs présents pour ces entités sont forcément pertinents.

Le deuxième algorithme, intitulé *Maxy*, sélectionne les n premières entités qui ont le plus d'attributs. L'intérêt de cette sélection est de favoriser les attributs des entités

10. Une liste d'attributs définie manuellement que nous jugeons inutiles.
11. Formulation RDF d'un système géodésique associé au GPS.
12. Moyen communautaire, léger, pour à étendre les informations géographiques.

surchargées et d'analyser leur impact sur la présentation des résultats. Nous adoptons ce choix car dans une entité qui contient beaucoup d'attributs, nous trouverons forcement des attributs représentatifs de la classe en question. L'algorithme est similaire à *Shorty*, il suffit de remplacer $\arg\min_{e_i \in C^*}(|e_i|)$ par $\arg\max_{e_i \in C}(|e_i|)$.

Finalement le troisième algorithme, intitulé *Fantasy*, sélectionne les n premières entités qui ont le plus de liens entrants ($interLink$) et liens sortants ($extLink$). Autrement dit, au niveau de chaque entité de la classe c, il existe des valeurs d'attributs, elles-mêmes de type entité, qui pointent vers d'autres entités et inversement (des entités qui pointent vers l'entité de la classe c à travers des valeurs d'attributs). L'intérêt de cette sélection est de favoriser les attributs des entités qui référencent le plus ou qui ont été le plus référencées par les éditeurs de Wikipedia. Nous pouvons ainsi analyser leur impact sur la présentation des résultats. En effet les résultats des évaluations effectuées par (Vercoustre *et al.*, 2007) ont démontré que la simple utilisation des liens structurés de Wikipedia donne de bonnes performances au niveau du classement des entités. Le calcul de cette sélection est similaire aux autres en remplaçant seulement $\arg\min_{e_i \in C^*}(|e_i|)$ par $\arg\max_{e_i \in C}(interLink + extLink)$.

Algorithme 1 Sélection des entités Shorty

soit $C = \{e_1, e_2, \ldots, e_z\}$ l'ensemble des entités qui appartiennent à la classe
$nbSelectedEntities \leftarrow n$
$Shorty \leftarrow \emptyset$
tant que $|Shorty| < nbSelectedEntities$ **faire**
    $S_{select} = \arg\min_{e_i \in C^*}(|e_i|)$ avec $C^* = \{e_i, |e_i| > 1\}$
    $Shorty \leftarrow S_{select}$
    $C^* \leftarrow C^* - \{S_{select}\}$
fin tant que

Après avoir filtré nos attributs et déterminé nos entités sélectionnées, nous pouvons passer au tri des attributs restant de la classe servant de requête et au calcul du score de chacun basé sur la sélection d'entités précédentes.

3.4. *Tri des attributs*

Nous devons, à ce niveau, trier les attributs en calculant un score $score(a)$ pour chaque attribut a de la classe c servant de requête afin d'identifier les attributs les plus pertinents pour l'agrégation. Sans cette étape, si nous agrégeons les attributs retournés sans les trier, nous risquons d'avoir un tableau troué dès la présentation des premiers attributs et les attributs les plus importants risquent d'être perdus et cachés dans l'ensemble. Le score $score(a)$ peut être calculé de trois manières.

– Score basé sur la fréquence (1ère formule)

En premier lieu, nous proposons d'utiliser la fréquence d'apparition des attributs dans la classe traitée ($tf\,(a,c)$) sur le nombre d'attributs dans la classe en question pour normaliser ($|c|$).

$$score\,(a) = \frac{tf\,(a,c)}{|c|} \qquad [1]$$

Pour cette formule nous ne prenons pas en compte le classement des entités et considérons que toutes les entités ont le même impact. Nous favorisons ainsi les attributs qui se répètent dans la majorité des entités et les affichons devant ceux qui se répètent rarement. De cette façon nous espérons avoir moins de cases vides dans le tableau à afficher.

– Score basé sur la probabilité de pertinence (2ème et 3ème formules)

Une autre façon de faire est de considérer le score $score\,(a)$ comme une probabilité d'appartenance $P\,(a|c)$ à la classe recherchée c pour chaque attribut traité a.
En traitant les attributs comme des termes, les entités comme des documents et la classe comme la collection traitée, nous pouvons faire une analogie avec le principe de classement probabiliste (*Probability Ranking Principle*), énoncé par (Robertson, 1997). Dans notre cas, le mieux est de retourner les attributs a et les entités e en ordre décroissant de leur probabilité sachant la classe c.
Dans la suite, nous allons utiliser la formule de probabilité totale de cette façon :

$$score\,(a) = P\,(a|c)\ = \sum_{e \in c} P\,(a|e)\ \cdot P\,(e|c) \qquad [2]$$

avec :

- $P\,(a|e)$: probabilité de pertinence de l'attribut a sachant l'entité e.
- $P\,(e|c)$: probabilité de pertinence de l'entité e sachant la classe c.

La probabilité $P\,(a|e)$ peut être calculée selon les deux formules suivantes :

- **Deuxième formule**

$P\,(a|e)$ est égale a l'inverse du nombre d'attributs $|e|$ dans l'entité e.

$$\begin{aligned} score\,(a) = P\,(a|c) \quad &= \textstyle\sum_{e \in c} \left(P\,(a|e)\ \cdot P\,(e|c)\right) \\ &= \textstyle\sum_{e \in c} \left(\frac{1}{|e|} \cdot P\,(e|c)\right) \end{aligned} \qquad [3]$$

Tous les attributs ont le même poids au sein de l'entité à laquelle ils appartiennent mais les attributs des entités ayant le moins d'attributs deviennent plus représentatifs que ceux des autres entités.

- **Troisième formule**

Pour cette troisième formule, nous utilisons le ratio de la fréquence $tf\,(a,c)$ de l'attribut a dans la classe c sur la somme des fréquences $tf\,(a_i,c)$ des attributs a_i de la classe c à laquelle appartient l'entité e en question. Ce qui donne comme formule fi-

nale :

$$score\,(a) = P\,(a|c) \quad \begin{array}{l} = \sum_{e \in c} \left(P\,(a|e,c) \cdot P\,(e|c) \right) \\ = \sum_{e \in c} \left(\frac{tf(a,c)}{\sum_{a_i \in e} tf(a_i,c)} \cdot P\,(e|c) \right) \end{array} \quad [4]$$

Cette formule combine l'intérêt des deux premières formules. En effet, elle donne plus d'importance aux attributs fréquents dans la classe et moins à ceux qui appartiennent aux entités possédant beaucoup d'attributs.

Selon les trois sélections déterminées dans la section 3.3, la probabilité $P\,(e|c)$ est calculée selon l'appartenance de l'entité e à la sélection traitée E. Dans notre étude, nous souhaitons que $P\,(e|c)$ pour $e \in E$ soit le double de $P\,(e|c)$ pour $e \notin E$ (afin de favoriser les attributs des entités de la sélection traitée).

Après calcul, ces valeurs sont estimées comme suit :

$$\begin{cases} P\,(e|c) = 2/(|c| + |E|) & si\ e \in \{E\} \\ P\,(e|c) = 1/(|c| + |E|) & sinon \end{cases} \quad [5]$$

avec

– E : l'ensemble des entités à valoriser qui peut être *Shorty, Maxy* ou *Fantasy*.

– $|c|$ et $|E|$: le nombre d'entités dans la classe c et la sélection traitée E

Les probabilités vérifient que $\sum_{e \in c} P(e|c) = 1$.

Une fois que les données sont collectées et triées pour une requête classe, nous les agrégeons dans un tableau.

3.5. *Agrégation des données*

Pour présenter les informations extraites et triées, nous adoptons la forme la plus adéquate et la plus lisible à notre sens pour l'utilisateur, celle des tableaux. Ces tableaux contiennent, pour une requête de type classe, dans chaque ligne une entité décrite par des attributs situés dans les colonnes.
Nous plaçons les cinq premières entités de la sélection Maxy dans les lignes et les dix attributs les plus importants dans les colonnes[13] (voir figure 6). Nous avons choisi la sélection Maxy car elle est censée être la sélection qui contient le plus d'attributs ; cela limitera donc le risque de présenter des cellules vides à l'utilisateur. Dans chaque cellule du tableau nous pouvons trouver soit une valeur atomique comme le nom, la langue, la date de naissance, soit une liste de valeurs par exemple les références. Une cellule peut également être aussi vide (sans valeur) si DBpedia n'a pas la valeur correspondante à cet attribut de l'entité en question.

13. Les valeurs 5 et 10 sont prises à titre expérimental.

	comment	Image	family	order	class	hasPhotoCollection	kingdom	genus	division	reference
Avocado46	The avocado (Persea americana), aguacate,...		Entité:Lauraceae	Entité:Laurales	Magnoliids Entité:Magnoliid	http://www4.wiwiss.fu-berlin.de/flickrwr...	Entité:Plant	Persea Entité:Persea		http://www.crfg.org/pubs/ff/avocado.html http://www.avocado.org.au/ http://avocadosource.com/ http://www.fao.org/es/ess/top/commodity.... http://www.avocados.es/ ...
Pineapple45	Pineapple (Ananas comosus) is the common....		Entité:Bromelioideae Entité:Bromeliaceae	Entité:Poales Entité:Commelinids	Entité:Monocots	http://www4.wiwiss.fu-berlin.de/flickrwr...	Entité:Plantae	Ananas Entité:Ananas	Entité:Angiosperms	https://scholarspace.manoa.hawaii.edu/bi... http://www.crfg.org/pubs/ff/pineapple.ht...
Eggplant44	The eggplant or aubergine (Solanum melon....		Entité:Solanaceae	Entité:Asterids Entité:Solanales	Entité:Eudicots	http://www4.wiwiss.fu-berlin.de/flickrwr...	Entité:Plantae	Solanum Entité:Solanum	Entité:Angiosperms	http://www.nhm.ac.uk/research-curation/p... http://www.nhm.ac.uk/research-curation/r... http://faostat.fao.org/site/567/DesktopD... http://www.learn2grow.com/plantdatabase/... http://www.pfaf.org/database/plants.php?...
Pomegranate44	A pomegranate is a fruit-bearing deciduo....		Entité:Lythraceae	Entité:Myrtales	Entité:Rosidae Entité:Magnoliopsida	http://www4.wiwiss.fu-berlin.de/flickrwr...	Entité:Plant	Entité:Punica Punica	Entité:Flowering_plant	http://www.pomegranate.me.uk/ http://www.crfg.org/pubs/ff/pomegranate.... http://www.jewishgiftplace.com/Symbolism... http://altmedicine.about.com/od/druginte... http://www.pomegranates.org/home.shtml
Banana42	Banana is the common name for herbaceous...		Entité:Musaceae	Entité:Zingiberales Entité:Commelinids	Entité:Monocots	http://www4.wiwiss.fu-berlin.de/flickrwr...	Entité:Plantae	Entité:Musa_%28genus%29	Entité:Angiosperms	http://www.iita.org/cms/details/banana_o... http://books.google.com/books?id=cSoYd-o... http://www.nytimes.com/2008/06/18/opinio... http://permaculture.org.au/2008/06/23/ba...

Figure 6. *Tableau résultat de la requête "Tropical_fruit"*

4. Evaluation

Afin d'évaluer notre approche, nous avons tout d'abord défini 30 requêtes (classes) qui sont soumises à DBpedia comme indiqué dans la section 3.1. Ces requêtes sont listées ci-dessous (voir figure 7). Nous les avons sélectionnées parmi les trois catégorisations Yago, Skos et OWL-DBpedia. Ces classes ont été choisies en s'appuyant sur des articles de Wikipedia. Les classes devaient également comporter entre 20 et 100 entités. Pour chaque classe soumise à DBpedia, nous avons extrait les entités, attributs et valeurs selon les étapes définies précédemment, puis les résultats sont présentés sous forme d'un tableau. Les 3 sélections d'entités (*Shorty, Maxy, Fantasy*) et les 3 formules de tri d'attributs proposées nous permettent de construire 9 tableaux, dont la description est donnée ci-après.
Pour la première formule (équation 1), nous construisons un seul tableau qui ne prend pas en compte les sélections d'entités que nous avons évoquées dans la section précédente. Pour la deuxième et la troisième formule (équations 3 et 4), nous calculons la probabilité $P\,(a|c)$ selon les trois sélections d'entités *Shorty, Maxy, Fantasy*, avec un quatrième cas qui ne prend pas en considération le classement des entités ($P(e|c)$ est toujours égal à 1). De cette façon, nous aurons quatre tableaux pour chaque probabilité. Nous obtenons donc à la fin neuf tableaux pour chaque requête c'est-à-dire pour chaque classe.

4.1. *Procédure d'évaluation*

Notre but est d'évaluer la qualité des tableaux construits. Il n'existe cependant pas de mesure spécifique pour estimer cette qualité. Nous avons donc décidé d'avoir recours aux jugements humains et aux mesures d'évaluation adoptées en RI classique.

Figure 7. *Liste des classes utilisées pour l'évaluation*

Pour ce faire, nous avons mis en place une interface appropriée[14]. Elle affiche pour chaque requête (classe) posée, d'abord, tous les attributs de la classe et ensuite les neuf tableaux correspondant aux différentes formules et sélections. A l'aide de cette interface, nous avons demandé à quatre volontaires de tester notre approche en donnant à chacun dix requêtes parmi les trente choisies antérieurement[15].
Chaque volontaire devait, pour chaque requête, cocher parmi les attributs affichés ceux qui lui semblaient pertinents pour cette dernière. Ensuite il devait sélectionner le tableau qui semblait le plus représentatif de son besoin.
Nous avons adopté dans notre étude deux types d'évaluation.
Nous nous sommes basés, tout d'abord, sur les observations des volontaires et leur jugement personnel (c'est-à-dire leur choix du meilleur tableau présenté). Nous avons aussi utilisé deux mesures d'évaluation qui sont la précision à dix (P@10) pour évaluer la capacité du système à afficher des attributs pertinents[16] et la précision moyenne ou MAP(*Mean Average Precision*) pour évaluer l'efficacité du système en terme de tri des attributs[17].

14. Cette interface, développée en Java, accède en ligne à DBPedia et construits les résultats à la volée.
15. Certaines requêtes ont été jugées en double.
16. Dans notre cas, la précision à 10 est le rapport entre le nombre d'attributs pertinents parmi les 10 premiers renvoyés.
17. La précision moyenne est la moyenne des valeurs de précision à chaque attribut pertinent de la liste ordonnée.

	Formule 1 (équation 1)	Formule 2 (équation 3)	Formule 3 (équation 4)
Pourcentage de sélection	7.5%	32.5%	60%

Tableau 1. *Choix des utilisateurs pour les formules sélectionnées*

	Sans sélection	Shorty	Maxy	Fantasy
Pourcentage de sélection	16.6%	8.3%	33.3%	41.8%

Tableau 2. *Choix des utilisateurs pour les différentes sélections d'entités au niveau de la troisième formule*

4.2. *Résultats*

Notre première évaluation tente de montrer les préférences (choix) des utilisateurs sur les tableaux renvoyés indépendamment de l'algorithme de sélection. La table 1 montre que 60% des utilisateurs choisissent les tableaux formés à partir de la troisième formule. La prise en compte de la fréquence de l'attribut dans la classe et l'entité est donc plus bénéfique.

Afin d'approfondir notre étude, nous nous intéressons maintenant aux sélections d'entités adoptées dans la troisième formule. Le tableau 2 nous montre que la sélection *Fantasy* avec ses 41,8% est la plus choisie par les utilisateurs suivie par la sélection *Maxy* avec 33,3% alors que *Shorty* n'a récolté que 8.3%.
Ceci signifie que les entités les plus importantes en terme de liens et celles qui ont le plus d'attributs permettent de favoriser l'apparition des attributs représentatifs. Par contre, les entités qui n'ont pas beaucoup d'attributs n'ont pas un véritable impact sur le classement des attributs.

Nous avons ensuite évalué la précision à 10 et la MAP pour la liste d'attributs renvoyés pour chacune des sélections et formules de tri. Le tableau 3 liste ces résultats. Nous constatons que l'approche est capable de renvoyer au moins cinq attributs pertinents parmi les dix (P@10>0,5). Cependant, la troisième formule offre le meilleur résultat avec 0,595 de P@10 en moyenne pour les sélections *Maxy* et *Fantasy* alors que la première formule ne permet d'obtenir que 0,53. Ces résultats confirment les observations faites par les usagers et nous montrent que la troisième formule permet d'obtenir de meilleures performances que la première.
Considérons maintenant les sélections. La sélection *Fantasy* associée aux deux dernières formules offre le meilleur tri des attributs avec 0,489 de MAP pour la troisième formule et 0,477 pour la deuxième formule. Cela soutient, à nouveau, les choix des usagers et montre que les entités importantes en termes de liens classent les attributs pertinents mieux que les autres. La MAP la plus faible est obtenue sans

		p@10	MAP
Formule 1 (équation 1)	Sans sélection	0,53	0,425
Formule 2 (équation 3)	Sans selection	0,58	0,473
	Shorty	0,57	0,47
	Maxy	0,54	0,456
	Fantasy	0,541	0,477
Formule 3 (équation 4)	Sans selection	0,58	0,37
	Shorty	0,583	0,45
	Maxy	0,595	0,475
	Fantasy	0,595	0,489

Tableau 3. *P@10 moyenne et MAP moyenne pour chaque formule et chaque sélection*

sélection des entités dans la majorité des formules, ce qui pourrait signifier que les entités ont un impact sur le classement des attributs.
La majorité des formules ont une MAP au-dessous de 0,5 même si P@10 permet d'obtenir relativement de bonnes performances. Une telle constatation nous conduit à conclure que les attributs pertinents sont bien parmi les dix premiers mais pas forcément bien placés dans les tableaux ce qui nous pousse à améliorer davantage notre approche.

En conclusion de ces évaluations préliminaires, nous avons montré que les résultats d'agrégations récupérés par notre approche satisfont globalement les utilisateurs. D'autres évaluations sont nécessaires pour valider toute l'approche. En particulier, nous envisageons de connecter l'approche à un moteur de recherche pour comparer ses résultats avec ceux sous forme de liste renvoyés par le moteur.

5. Conclusion et perspectives

Le travail présenté dans cet article s'inscrit dans le cadre de la recherche agrégée. Nous avons élaboré une approche qui génère, pour une classe, un groupe d'entités homogènes et ses attributs avec des valeurs associées collectés à partir de la base de connaissance DBpedia.
Pour mettre en œuvre cette approche nous avons proposé plusieurs formules. L'intérêt de ces formules est de mesurer l'importance de chaque attribut par rapport à la classe traitée. Nous avons cherché à mettre en relief les attributs les plus représentatifs de la classe. Afin de mesurer l'impact des entités sur les attributs, nous avons sélectionné trois groupes d'entités *Shorty*, *Maxy* et *Fantasy* et nous les avons introduites dans nos formules.
Pour présenter nos résultats aux utilisateurs, nous avons adopté une structure lisible à savoir le tableau. Avec l'aide de 4 personnes, nous avons effectué une évaluation manuelle de notre approche, ce qui nous a permis de calculer plusieurs mesures

d'évaluation traditionnelles (P@10 et MAP).
L'approche proposée a l'avantage de traiter globalement plusieurs problèmes. En effet, cette approche est capable de percevoir à peu près les attentes des utilisateurs et leur offre jusqu'à 6 attributs pertinents dans les tableaux (la meilleure P@10 est de 0,595).

Le travail réalisé dans cet article ouvre diverses perspectives à court terme comme estimer l'impact du nombre d'entités d'une classe sur le résultat, implémenter la recherche à partir de requêtes entité, évaluer d'autres formules pour le calcul de pertinence des attributs et étendre le nombre de requêtes et le nombre d'usagers pour l'évaluation. A plus long terme, il faudrait également mettre en place un module de filtrage et vérification de valeurs des cellules du tableau et exploiter les ontologies reliées a DBpedia comme Yago et OWL-DBpedia pour améliorer la qualité et la lisibilité des données représentées dans les tableaux.

6. Bibliographie

Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z. G., « DBpedia : A Nucleus for a Web of Open Data », *Proceedings of ISWC/ASWC*, p. 722-735, 2007.

Bizer C., Lehmann J., Kobilarov G., Auer S., Becker C., Cyganiak R., Hellmann S., « DBpedia - A crystallization point for the Web of Data », *Journal of Web Semantics : Science, Services and Agents on the World Wide Web*, vol. 7, n° 3, p. 154-165, 2009.

Boughanem M., Savoy J., *Recherche d 'information. Etat des lieux et perspectives*, Lavoisier, 2008.

Cafarella M. J., Halevy A. Y., Zhang Y., Wang D. Z., 0002 E. W., « Uncovering the Relational Web », *Proceedings of WebDB*, 2008.

Elmeleegy H., Madhavan J., Halevy A. Y., « Harvesting Relational Tables from Lists on the Web », *Proceedings of VLDB*, vol. 2, n° 1, p. 1078-1089, 2009.

Kato M. P., Ohshima H., Oyama S., Tanaka K., « Query by analogical example : relational search using web search engine indices », *Proceedings of CIKM*, p. 27-36, 2009.

Medelyan O., Milne D. N., Legg C., Witten I. H., « Mining meaning from Wikipedia », *International Journal of Human-Computer Interactions*, vol. 67, n° 9, p. 716-754, 2009.

Murdock V., Lalmas M., « Workshop on aggregated search », *SIGIR Forum*, vol. 42, n° 2, p. 80-83, 2008.

Paris C., Wan S., Thomas P., « Focused and aggregated search : a perspective from natural language generation », *Information Retrieval*, vol. 13, n° 5, p. 434-459, 2010.

Robertson S. E., *The probability ranking principle in IR*, Readings in information retrieval, Morgan Kaufmann Publishers Inc., p. 281–286, 1997.

Sauper C., Barzilay R., « Automatically Generating Wikipedia Articles : A Structure-Aware Approach », *Proceedings of ACL 2009*, 2009.

Suchanek F. M., Kasneci G., Weikum G., « Yago : a core of semantic knowledge », *Proceedings of WWW Conference*, p. 697-706, 2007.

Vercoustre A.-M., Pehcevski J., Thom J. A., « Using Wikipedia Categories and Links in Entity Ranking », *Proceedings of INEX*, p. 321-335, 2007.