Une approche pour la recherche sémantique de l'information dans les documents semi-structurés hétérogènes

Yassine Mrabet, Nacéra Bennacer Seghouani, Nathalie Pernelle, Mouhamadou Thiam

Une approche pour la recherche sémantique de l'information dans les documents semi-structurés hétérogènes

Yassine Mrabet [1,2] — Nacéra Bennacer [2] — Nathalie Pernelle [1] — Mouhamadou Thiam [1,2]

1 LRI, Université Paris-Sud 11, INRIA Saclay
F-91893 Orsay Cedex, France
2 SUPELEC Sciences des Systèmes (E3S) Département Informatique
91192 Gif-sur-Yvette Cedex, France
1 {Prenom.Nom}@lri.fr 2 {Prenom.Nom}@supelec.fr

RÉSUMÉ. *Ce papier présente SHIRI-Querying, une approche pour la recherche sémantique de l'information dans les documents semi-structurés. Nous proposons une solution pour pallier l'incomplétude et l'imprécision des annotations au moment de l'interrogation. Cette solution repose sur deux types de reformulations élémentaires qui exploitent la notion d'agrégation et la structure des documents. Nous présentons l'algorithme $DREQ$ qui combine ces transformations élémentaires pour construire des reformulations ordonnées de la requête utilisateur. L'étude de notre approche sur deux corpus réels montre que les reformulations augmentent considérablement le rappel et que la précision est meilleure pour les premières réponses retournées.*

ABSTRACT. *This paper presents* SHIRI-Querying*, an approach for semantic search in semi-structured documents. We propose a solution to tackle incompleteness and imprecision of annotations at querying time. This solution relies on two elementary reformulation types that use the notion of aggregation and the documents structure. We present the dynamic algorithm ($DREQ$) which combines these elementary transformations to construct ordered reformulations of user queries. Experimentations on two real datasets show that reformulations increase greatly the recall and that the precision is better for the first returned answers.*

MOTS-CLÉS : *Ontologies, Web sémantique, SPARQL/RDF(S)/OWL, Reformulations de requêtes*

KEYWORDS: *Ontologies, Semantic Web, SPARQL/RDF(S)/OWL, Query Reformulations*

1. Introduction

La recherche sémantique de l'information est une des principales motivations du Web sémantique. Un moteur de recherche sémantique peut être vu comme un outil qui répond à des requêtes – formulées avec les concepts et les relations d'une ontologie de domaine – en les alignant avec des annotations sémantiques des documents cibles. Dans une vue idéale, ce problème de Recherche d'Information (RI) peut être considéré comme similaire à la RI dans les bases de données relationnelles où les réponses sont des ensembles de tuples satisfaisant la requête de l'utilisateur. Cependant, cette vue ne se concrétise que si les contenus de tous les documents peuvent être représentés par des instances de concepts ou de relations définis dans une ontologie donnée.

Les avancées de la recherche visant à automatiser le peuplement des ontologies et l'annotation des documents sont prometteuses (Popov *et al.,* 2004, Cimiano *et al.,* 2005, Etzioni *et al.,* 2005, Thiam *et al.,* 2009). Cependant, la localisation précise de toutes les instances dans un document reste une tâche difficile. Une certaine imprécision sémantique peut se produire du fait que les métadonnées choisies ne sont pas parfaitement appropriées (e.g. utiliser le concept « Événement » au lieu du concept « Conférence »). De plus, les annotations sont souvent incomplètes. C'est plus particulièrement le cas des relations sémantiques qui sont difficiles à trouver quand les documents sont hétérogènes.

Dans (Hurtado *et al.,* 2006, Corby *et al.,* 2006), les auteurs remédient á l'imprécision sémantique de l'annotation en appliquant des approximations guidées par une ontologie de domaine (e.g. en exploitant la subsomption, la proximité contextuelle ou des chemins de relations sémantiques). Cependant, ces approches ne traitent pas les cas où l'annotateur a localisé imprécisément les instances dans les documents et ne s'intéressent pas au problème d'incomplétude des annotations. D'autres travaux (Castells *et al.,* 2007, Bhagdev *et al.,* 2008) combinent la recherche d'information guidée par des ontologies et la RI mots clés afin de pallier l'incomplétude des annotations sémantiques. Cette recherche hybride permet d'augmenter le rappel mais, en contre partie, un ensemble de contraintes sémantiques est ignoré.

Dans cet article, nous proposons une approche de recherche d'information sémantique appelée *SHIRI-Querying*. Les contributions de l'approche sont : (1) une méthode de reformulation de requêtes permettant d'interroger des annotations sémantiques incomplètes et imprécises (2) une relation d'ordre entre les différentes reformulations permettant de privilégier les requêtes les plus sémantiques et (3) un algorithme dynamique, appelé ($DREQ$), qui reformule la requête de l'utilisateur et exécute les reformulations suivant la relation d'ordre définie.

SHIRI-Querying utilise le modèle d'annotation générique du système *SHIRI*[1] décrit en OWL (W3C). Ce modèle décrit (1) l'ontologie de domaine et (2) des métadonnées spécifiques à l'annotation. Grâce à un adaptateur, nous transformons les annotations produites par différents annotateurs sémantiques en des annotations RDF

1. SHIRI : projet Digiteo labs (LRI, SUPELEC)

conformes au modèle d'annotation. La granularité de ces annotations est le nœud de document (e.g. balise HTML ou XML). Chaque nœud est annoté comme contenant une ou plusieurs instances de (différents) concepts de domaine. Le modèle d'annotation utilisé permet aussi de représenter les liens de structure entre les nœuds. Ces liens sont exploités en complément des autres métadonnées du modèle afin de trouver des réponses quand les relations sémantiques requises ne sont pas annotées.

SHIRI-Querying prend en entrée des requêtes utilisateur écrites en langage SPARQL (W3C), formulées à l'aide des concepts et des relations définies dans une ontologie de domaine. Ces requêtes sont reformulées en utilisant deux types de reformulations élémentaires : (i) la *reformulation par agrégation* qui permet de retrouver des instances imprécisément annotées et (ii) la *reformulation par voisinage* qui permet de localiser des instances dans des nœuds proches dans le document.

L'algorithme *DREQ* combine ces transformations élémentaires pour construire dynamiquement les reformulations et les exécuter suivant une relation d'ordre définie. Cette relation d'ordre privilégie les réponses où les nœuds de documents sont précisément annotés et liés par les relations sémantiques requises. Contrairement aux approches qui ont besoin de travailler sur les réponses et/ou l'ensemble des annotations, dans *SHIRI-Querying*, les réponses sont triées de facto puisque la construction des reformulations est ordonnée.

Dans cet article, nous commençons par présenter un bref état de l'art. Dans la section 3, nous présentons brièvement le système *SHIRI-Querying*, son architecture et le modèle d'annotation. Dans la section 4, nous présentons les deux types de reformulations élémentaires et l'algorithme $DREQ$ qui permet de les combiner. La section 5 résume les résultats de nos expérimentations effectuées sur deux corpus réels. Enfin, dans la section 6, nous concluons et donnons quelques perspectives.

2. État de l'art

Plusieurs approches de recherche sémantique (Hurtado *et al.,* 2006, Corby *et al.,* 2006) approximent les concepts et les relations de la requête utilisateur en utilisant l'ontologie de domaine (e.g. en exploitant la subsomption, les chemins de relation, la proximité contextuelle). Ces approches s'attaquent à l'imprécision sémantique des annotations alors que nous essayons d'adapter les requêtes de l'utilisateur à des annotations incomplètes incluant potentiellement des imprécisions dans la localisation des instances dans les documents.

D'un autre côté, certaines approches hybrides (Castells *et al.,* 2007, Bhagdev *et al.,* 2008) s'attaquent à l'incomplétude des annotations sémantiques en combinant la recherche mots-clés et la recherche sémantique. Dans (Bhagdev *et al.,* 2008), les résultats de la requête utilisateur sont les documents appartenant à l'intersection des résultats de la partie mots-clés et des résultats de la partie sémantique de la requête. Dans (Castells *et al.,* 2007), les auteurs proposent un modèle de recherche sémantique basé sur la modélisation classique en vecteur de poids et la granularité des annota-

tions sémantiques est le document entier. Cependant, les approches qui utilisent les mots-clés comme support à la recherche sémantique éliminent complètement certaines contraintes sémantiques de la requête. Dans notre cas, nous conservons des contraintes sémantiques tout au long du processus de reformulation. Ces deux types d'approches (hybrides et d'approximation sémantique) pourraient être utilisés conjointement à la notre pour améliorer le rappel.

D'autres approches utilisent la structure des documents pour répondre à la requête des utilisateurs. Dans (Hristidis *et al.,* 2006, N appil a *et al.,* 2007), les requêtes traitées sont des ensembles de noms de balises XML, et la proximité structurelle des nœuds est exploitée pour retourner les balises recherchées. Par exemple, (N appil a *et al.,* 2007) utilise la structure des documents XML pour construire des graphes connexes dont les nœuds sont les balises requises par l'utilisateur, définissant ainsi des "plus petits contextes" relativement à la structure des documents et aux nœuds requis. Cependant, ces approches restent non sémantiques et ne sont pas adaptées aux documents hétérogènes sans structure régulière où chaque nœud peut contenir des instances de différents concepts.

3. L'approche SHIRI-Querying

3.1. *Brève description de l'architecture*

SHIRI-Querying (cf. figure 1) utilise les standards W3C RDF/OWL pour la représentation des ressources et SPARQL pour leur interrogation. L'*adaptateur* utilise un ensemble de règles pour rendre les annotations sémantiques produites par les outils tiers conformes au modèle d'annotation. Le *moteur de requêtes* utilise les reformulations élémentaires pour répondre aux requêtes utilisateur.

3.2. *Description du modèle d'annotation*

Soit $\mathcal{O}(\mathcal{C}_\mathcal{O}, \mathcal{R}_\mathcal{O}, \preceq, \mathcal{D}_\mathcal{O}, \mathcal{X}_\mathcal{O})$ une ontologie de domaine où $\mathcal{C}_\mathcal{O}$ est l'ensemble des concepts, $\mathcal{R}_\mathcal{O}$ est l'ensemble des relations entre les concepts, $\preceq$ dénote la relation de subsomption entre concepts ou relations, $\mathcal{D}_\mathcal{O}$ définit le domaine et le range de chaque relation et $\mathcal{X}_\mathcal{O}$ est un ensemble d'axiomes.

Le modèle d'annotation $\mathcal{A}$ est généré automatiquement depuis l'ontologie de domaine. $\mathcal{A}$ est défini par le quintuplet $(\mathcal{C}_\mathcal{A}, \mathcal{R}_\mathcal{A}, \preceq, \mathcal{D}_\mathcal{A}, \mathcal{X}_\mathcal{A})$ où $\mathcal{C}_\mathcal{A} = \mathcal{C}_\mathcal{O} \cup \mathcal{C}_\mathcal{S}$, $\mathcal{R}_\mathcal{A} = \mathcal{R}_\mathcal{O} \cup \mathcal{R}_\mathcal{S}$. $\mathcal{C}_\mathcal{S}$ et $\mathcal{R}_\mathcal{S}$ sont les concepts et les relations définies pour la tâche d'annotation. $\mathcal{X}_\mathcal{A} = \mathcal{X}_\mathcal{O} \cup \mathcal{X}_\mathcal{S}$ où $\mathcal{X}_\mathcal{S}$ est l'ensemble des règles utilisées par l'adaptateur pour la mise en conformité des annotations de domaine. Dans ce modèle (cf. figure 2), les instances de concepts de $\mathcal{C}_\mathcal{A}$ sont identifiées par les URIs des nœuds de document et les littéraux associés par la relation *hasValue* correspondent aux contenus textuels de chacun des nœuds. Nous définissons dans $\mathcal{C}_\mathcal{S}$ et $\mathcal{R}_\mathcal{S}$ les métadonnées d'agrégation suivantes :

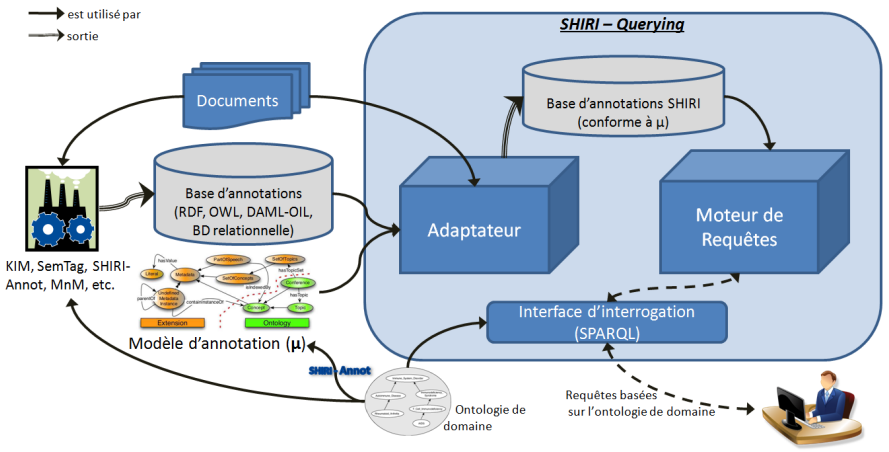


Figure 1. *Architecture de SHIRI Querying*

– Le concept *PartOfSpeech* est utilisé pour annoter un nœud de document contenant plusieurs instances de différents concepts (e.g. documents 2 et 3 dans la figure 3). Ces concepts seront ensuite référencés par la propriété *isIndexedBy*.

– La métadonnée *SetOfConcepts* est utilisée pour annoter les nœuds de document contenant plusieurs instances du même concept. Un concept $SetOfc_i$ est défini comme sous classe de *SetOfConcepts* pour chaque concept $c_i \in \mathcal{O}$ (e.g. $SetOfPersons$ dans le document 1 de la figure 3). Par ailleurs, nous définissons des relations $rSet$ et $rSet^{-1}$ dans $\mathcal{R}_\mathcal{S}$, dérivées de relations (de domaine) $r \in \mathcal{R}_\mathcal{O}$ afin de représenter les relations sémantiques potentielles entre une instance et un ensemble d'instances.

– La relation *neighborOf* est aussi définie dans $\mathcal{R}_\mathcal{S}$ pour représenter la proximité des nœuds dans le document. Elle pourra être définie en fonction des corpus.

Dans le cas où un nœud de document contient une seule instance de concept de domaine C (e.g. la conférence WWW2008 dans le document 1 de la figure 3), le nœud est annoté par C. Les attributs (propriétés littérales) de l'instance deviennent des attributs du nœud. Ces différentes métadonnées d'annotation des nœuds sont générées par l'adaptateur de SHIRI-Querying.

3.3. *Un cas d'utilisation*

La figure 3 illustre un exemple d'interrogation de 3 documents annotés décrivant des références bibliographiques. Les annotations sémantiques (e.g. instances des concepts *Person, Topic, Event* et *Article*) sont fournies par différents annotateurs.

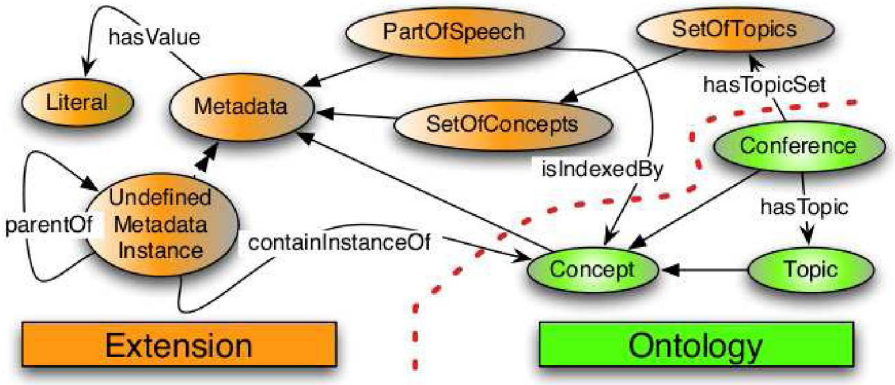


Figure 2. *Modèle d'Annotation*

L'adaptateur se charge ensuite de mettre ces annotations en conformité avec le modèle d'annotation.

Dans cet exemple le moteur de requêtes reformule une requête utilisateur recherchant les articles publiés dans la conférence WWW 2008 et leurs auteurs. Cette reformulation permet, entre autres, d'atteindre : (1) Les nœuds voisins annotés par les métadonnées *Event*, *SetOfPersons Article* et contenant le terme "WWW 2008" (nœuds du doc-1) (2) le nœud annoté par *PartOfSpeech*, indexé par *Event*, *Person* et *Article* et contenant le terme "WWW 2008" (nœud du doc-2) et (3) Les nœuds *PartOfSpeech* voisins dans le document et indexés par les concepts recherchés (nœuds du doc-3).

4. Reformulation des requêtes

Dans cette section, nous présentons dans un premier temps deux types de transformations élémentaires (par voisinage et par agrégation). Dans un second temps, nous présentons la relation d'ordre suivant laquelle les reformulations sont construites et l'algorithme $DREQ$ qui combine les transformations élémentaires pour reformuler la requête utilisateur.

4.1. *Définitions préliminaires*

Nous introduisons d'abord quelques notions préliminaires (W3C).

Graphe RDF. Considérons les ensembles infinis deux à deux disjoints I, B, et L (IRIs, Nœuds blancs et Littéraux). Un graphe RDF est un ensemble de triplets RDF $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ où s est le sujet, p le prédicat et o l'objet.

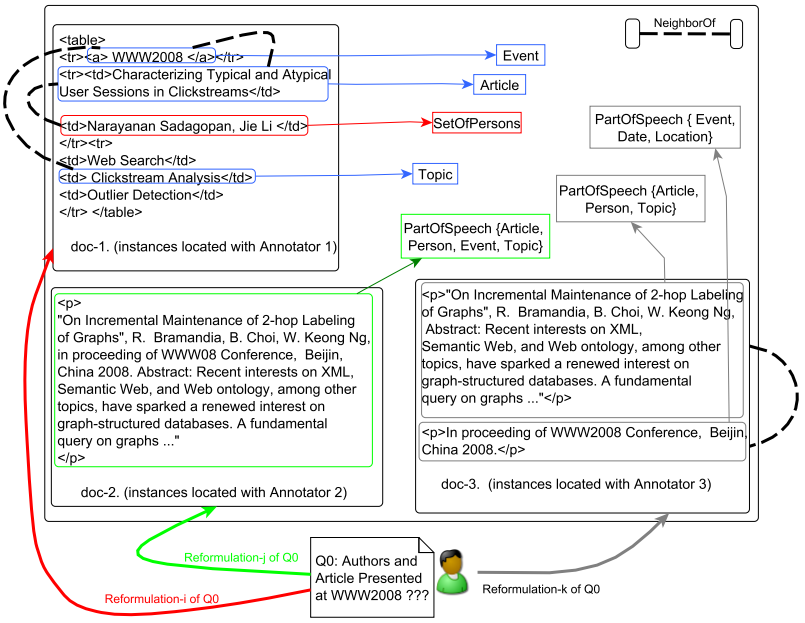


Figure 3. *Un cas d'utilisation*

Patron de graphe élémentaire. Considérons l'ensemble des variables V disjoint avec les ensembles I, B et L. Un patron de triplet est un triplet $(s, p, o) \in (I \cup V) \times (I \cup V) \times (I \cup V \cup L)$. Un patron de graphe élémentaire P est un ensemble de patrons de triplets. Un point d'interrogation $?v$ dans un triplet indique que v est une variable.

Une requête RDF est souvent vue comme la combinaison ensembliste de patrons de graphe élémentaires (e.g. union, intersection). Par souci de simplification, nous considérons uniquement les patrons de graphes élémentaires. Dans le cadre de notre approche, nous traitons des requêtes basées sur une ontologie que nous formalisons comme suit :

Requête basée sur une ontologie. Une requête q basée sur une ontologie Ω est définie par le quadruplet (P, S, F, D) où :

– P est un patron de graphe élémentaire conforme à Ω. Si Ω est l'ontologie de domaine $\mathcal{O}$, la requête est dite $\mathcal{O}$-conforme. Si Ω est le modèle d'annotation $\mathcal{A}$, la requête est dite $\mathcal{A}$-conforme. $V(P)$ dénote l'ensemble des variables qui sont utilisées dans P et $C(P)$ dénote l'ensemble des concepts utilisés dans la requête.

– F est un ensemble de contraintes définies par la combinaison logique d'expressions booléennes e. e_v dénote une expression de F utilisant la variable v.

– S est l'ensemble des variables sélectionnées (apparaissant dans la clause SELECT de la requête).

– D est le graphe RDF des annotations conformes au modèle $\mathcal{A}$. La réponse à la requête se fera en alignant P et F avec D.

Exemple 1. La requête utilisateur, $\mathcal{O}$-conforme, q_0 formulée en figure 3 est définie par (P_0,F_0,S_0,D) tel que :

P_0 : { $(?art, rdf:type, Article)$, $(?aut, rdf:type, Person)$, $(?aut, hasName, ?aName)$, $(?conf, rdf:type, Conference)$, $(?art, publishedIn, ?conf)$ $(?art, authoredBy, ?aut)$, $(?conf, hasName, ?cName)$}

F_0 : $?cName = "WWW2008"$

S_0 : { $?art$, $?aut$, $?aName$ }

D : est l'ensemble des triplets annotant les documents 1, 2 et 3.

Dans la figure 3, la requête q_0 n'a pas de réponse : les plateformes d'annotation n'ont pas retrouvé de relations sémantiques entre les instances. Le but de la reformulation de q_0 est d'exploiter les annotations générées par l'adaptateur pour répondre autrement à la requête utilisateur.

4.2. *Reformulation par voisinage*

Détecter les relations sémantiques dans les documents hétérogènes est une tâche difficile. Le but de la reformulation par voisinage est d'atteindre les nœuds qui pourraient être reliés par les relations sémantiques requises par l'utilisateur. En effet, nous supposons que les nœuds de documents structurellement proches sont probablement liés par des relations sémantiques. La reformulation par voisinage, noté f_v, remplace une relation sémantique de domaine par la relation $neighborOf$.

Définition 1. Soit q une requête $\mathcal{A}$-conforme définie par (P, F, S, D) et t un patron de triplet $(?s, r, ?o) \in P$ tel que $r \in \mathcal{R}_O$, alors $f_v(q, t)$ est une requête $\mathcal{A}$-conforme q' définie par (P', F, S, D) tel que $P' = P \backslash \{t\} \cup \{(?s, neighborOf, ?o)\}$.

Exemple 2. Si nous appliquons f_v au triplet $t = (?art, authoredBy, ?aut)$ de q_0, nous obtenons la requête $q'_1 = f_v(q_0, t)$ définie par (P', F, S, D) tel que :

P' : { $(?art, rdf:type, Article)$, ...$(?art, neighborOf, ?aut)$, ... } F, S, D : restent inchangés.

La reformulation par voisinage peut ainsi générer ce que nous appelons des patrons de sous-graphes sémantiquement indépendants.

Patron de sous-graphe sémantiquement indépendant. Un sous-graphe p de P est sémantiquement indépendant si :

– $\forall v_1, v_2 \in V(p)$ $(?v_1, r, ?v_2) \in P \rightarrow (?v_1, r, ?v_2) \in p$

– $\forall (?v_1, r, ?v_2) \in P$ et v_1 (resp. v_2) $\in V(p)$ et v_2 (resp. v_1) $\notin V(p)$, alors $r = neighborOf$

La notion de sous-graphes indépendants va nous permettre d'éviter les redondances dans la construction des reformulations de la requête utilisateur (cf. plan de construction des reformulations).

4.3. *Reformulation par agrégation*

La reformulation par agrégation exploite les métadonnées agrégées (i.e. les concepts *PartOfSpeech* et *SetOfConcept*) afin d'atteindre des nœuds de documents hétérogènes. De ce fait, la reformulation par agrégation est soit une *reformulation en ensemble*, soit une *reformulation en partie de discours*.

La *reformulation en ensemble*, notée f_e, suppose qu'une relation r recherchée entre deux instances de domaine de type c et c' peut être reformulée en une relation $rSet$ ou $rSet^-1$ entre deux instances de types c et $SetOfc'$ ou deux instances de types $SetOfc$ et c'. La *reformulation en partie de discours* suppose que les instances sémantiques requises peuvent être retrouvées dans des nœuds de documents correspondants à des parties de discours (i.e. annotés par $PartOfSpeech$). De même, cette reformulation se base sur le fait que les relations sémantiques requises peuvent exister entre les instances appartenant au même nœud $PartOfSpeech$.

Dans ce qui suit, nous détaillons la reformulation en partie de discours, notée f_{pdd}. Elle s'applique aux requêtes $\mathcal{A}$-conformes relativement à un sous graphe sémantiquement indépendant donné. Il s'agit dans une première phase de reporter tous les filtres littéraux au niveau du nœud de document. Rechercher une conférence dont le nom est "WWW 2008" revient alors à rechercher un nœud $PartOfSpeech$ indexé par le concept $Conference$ et contenant le terme "WWW 2008" . Ici, les contraintes d'égalité initiales sont juste remplacées par des contraintes d'inclusion, mais des mesures plus sophistiquées pourraient être employées. Afin de ne construire chaque requête qu'une seule fois (conformément au plan de construction détaillé ultérieurement), le sous graphe cible de la transformation doit être $\mathcal{O}$-conforme (i.e. ne contenant que des métadonnées de domaine).

Définition 2. Soit q une requête $\mathcal{A}$-conforme définie par (P, F, S, D) et soit p un sous graphe de P, sémantiquement indépendant et $\mathcal{O}$-conforme. La reformulation $f_{pdd}(q, p)$ est alors une requête $\mathcal{A}$-conforme q' définie par (P', F', S', D) tel que :
P' = $\{t \in P/ (?v_1, neighborOf, ?v_2)$ tel que v_1 (resp. v_2) $\in V(p)$ et v_2 (resp. v_1)$\notin V(p)$ est remplacé par $(?pos, neighborOf, ?v_2)$ (resp. $(?v_1, neighborOf, ?pos)$) et $(?v, attribute, ?l)$ t.q. $v \in V(p)$ et $?l$ de type littéral est remplacé par $(?pos, hasValue, ?lpos)\}$
$\cup\{(?pos, rdf : type, PartOfSpeech)\} \cup \{(?pos, indexedBy, c)/c \in C(p)\}$
$\cup \{(?pos, hasValue, lpos)\} - p$
$F' = \{e \in F/(?l = literal)$ or $(?l\ contains\ literal)$ s.t. $l \in V(p)$, est remplacé par $(?lpos\ contains\ literal)\ \}$
Si $S \cap V(p) \neq \phi$ alors $S' = S - V(p) \cup \{?pos\}$ sinon $S' = S$

Exemple 3. Si nous appliquons f_{pdd} à la requête q'_1 obtenu dans l'exemple 2, suivant le sous graphe sémantiquement indépendant $p = \{(?art,\ rdf : type,\ Article),\ (?conf, rdf : type, Conference),\ (?conf,\ hasName,\ ?cName),\ (?art,\ publishedIn,\ ?conf)\}$, nous obtenons la requête $q'_2 = f_{pdd}(q_0, p)$ définie par (P', F', S', D) tel que :

P' = { $(?pos, rdf : type, PartOfSpeech)$,
$(?pos, isIndexedBy, Conference)$, $(?pos, isIndexedBy, Article)$,
$(?pos, hasValue, ?lPos)$,$(?pos, neighborOf, ?aut)$,
$(?aut, rdf : type, Person)$, $(?aut, hasName, ?aName)$ }
F' = { $(lPos\ contains\ "WWW2008")$ }, et S' = { $?aut,\ ?aName, ?pos$ }

4.4. *Plan de construction des reformulations*

La reformulation d'une requête $q_0(P_0, F_0, S_0, D)$ est une requête $q_i(P_i, F_i, S_i, D)$ obtenue par la composition de reformulations par voisinage et par agrégation, suivant la relation d'ordre définie ci-dessous.

Relation d'ordre $\preccurlyeq$. Dans notre approche, nous considérons que les nœuds réponses sont plus pertinents s'ils ne contiennent pas d'instances agrégées et s'ils sont reliés par les relations sémantiques requises. Par exemple, les reformulations du triplet $\{<?art, authoredBy, ?aut>\}$ sont générées dans l'ordre suivant :

ordre 1 : {< ?art, authoredBySet, ?setOfPersons>} et
{< ?setOfArt, authorBySet^{-1}, ?aut>}
ordre 2 : {< ?pos, isIndexedBy, Person>,
< ?pos, isIndexedBy, Article>}
ordre 3 : {< ?art,neighborOf, ?aut>}
ordre 4 : {< ?SetOfArticle, neighborOf, ?aut>}
and {< ?art, neighborOf, ?setOfPersons>}
... :
ordre 8 : {< ?pos1, neighborOf, ?pos2>,
< ?pos1, isIndexedBy, Person>,
< ?pos2, isIndexedBy, Article>}

Ce raisonnement est généralisé pour tous les triplets d'une requête donnée.

Définition 3. En considérant $N(q)$, $Pos(q)$, $Sets(q)$ comme étant respectivement : le nombre de relations $neighborOf$, le nombre de $PartOfSpeech$ et le nombre de métadonnées $SetOfc$ utilisés dans la requête q, la relation d'ordre $\preccurlyeq$ entre deux reformulations quelconques q_i et q_j d'une même requête utilisateur est définie comme suit :

$q_i \preccurlyeq q_j$ ssi $(N(q_i) > N(q_j)) \vee (((N(q_i) = N(q_j)) \wedge$
$(Pos(q_i) > Pos(q_j))) \vee ((Pos(q_i) = Pos(q_j)) \wedge (Sets(q_i) \geq Sets(q_j))))$

$DREQ$ **: (Dynamic Reformulation and Execution of Queries algorithm)**
$DREQ$ permet de construire et d'exécuter les requêtes reformulées suivant la relation d'ordre $\preccurlyeq$. La complexité de construction des reformulations est en $O(e^{n^2})$, n étant le nombre de variables dans la requête utilisateur originale. Cependant, l'algorithme est dynamique et produit de nouveaux sous ensembles de reformulations (de même ordre) en temps polynomial. L'algorithme peut être stoppé à un seuil donné suivant la relation d'ordre $\preccurlyeq$, assurant ainsi que les réponses retrouvées sont celles produites par les meilleures reformulations. L'exécution des requêtes quant à elle dépend du moteur RDF utilisé. Nous utilisons CORESE(Corby *et al.,* 2006) pour interroger les annotations RDF.

Nous notons $F_v^i(q)$, l'ensemble des requêtes obtenues en appliquant f_v à i triplets distincts de la requête q. $F_e^j(q)$ dénote l'ensemble des requêtes obtenues en appliquant f_e à j variables distinctes de la requête q. $F_{pdd}^l(q)$ dénote l'ensemble des requêtes obtenues en appliquant f_{pdd} à l sous-graphes sémantiquement indépendants $p_1, ...p_l$ de la requête q.

5. Expérimentations

5.1. *Critères d'évaluation*

L'évaluation des systèmes de recherche sémantiques n'est pas une tâche évidente. Nous précisons dans ce qui suit les entrées et sorties du système ainsi que les critères de jugement des réponses. Les entrées du système sont des requêtes SPARQL basées sur une ontologie de domaine. Les données sont des annotations RDF de documents XML et/ou HTML. Les réponses proposées aux requêtes sont des faits RDF extraits depuis les annotations fournies par les outils tiers ou des faits extraits des annotations complémentaires sur les nœuds de documents (construites par l'adaptateur SHIRI). Dans ce dernier cas les textes des nœuds sont aussi montrés à l'utilisateur grâce à l'attribut *hasValue*. // Une réponse est jugée correcte si les faits/nœuds retournés correspondent/contiennent effectivement les instances de concepts recherchés par l'utilisateur et que ces instances sont reliées par les relations sémantiques exprimées dans la requête. Par exemple, si l'utilisateur recherche la conférence CORIA'2009 et sa date, un ensemble de nœuds contenant la bonne conférence mais la date d'un de ses workshops sera considéré comme fausse réponse.

Dans le cadre de ces expérimentations, la relation $neighborOf$ est définie comme un chemin non orienté de longueur $\leq d$ dans l'arbre XML ou HTML. Nous étudions en particulier comment la relation d'ordre $\preccurlyeq$ et la distance d influencent les résultats. Comme nous souhaitons uniquement évaluer les performances de l'approche de reformulation, nous prenons seulement en compte les annotations sémantiques correctes, i.e. les réponses comportant des bruits – générés par les systèmes d'annotation tiers – ne sont pas prises en compte.

Algorithme 1 DREQ ($q_0(P,\ F,\ S,\ D)$)

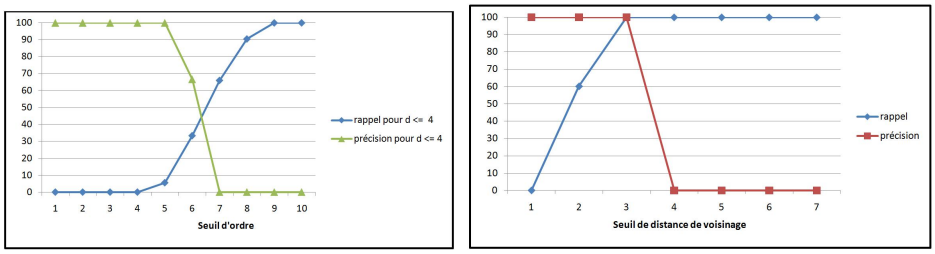
```
1  Variable
2      I : Entier  { nombre maximum de reformulations f_v possible }
3      J : Entier  { nombre maximum de reformulations f_e possible }
4      K : Entier  { nombre maximum de reformulations f_pdd possible }
5  Début
6      { 1ères reform. par agrégation (sans neighborOf), une seule f_pdd possible }
7      Pour j ∈ J Faire
8          générer et exécuter F_e^j(q_0)
9      générer et exécuter f_pdd(q_0)
10     { Combinaison des reformulations par voisinage et par agrégation }
11     Pour i ∈ I Faire
12         générer et exécuter F_v^i(q_0)
13         Pour j ∈ J Faire
14             Pour q ∈ F_v^i(q_0) Faire
15                 générer et exécuter F_e^j(q)
16         Pour k ∈ K Faire
17             Pour q ∈ F_v^i(q_0) Faire
18                 générer et exécuter F_pdd^k(q_0)
19             Pour j ∈ J Faire
20                 Pour q ∈ F_pdd^k(q_0) Faire
21                     générer et exécuter F_e^j(q)
22 Fin
```

5.2. *Premier corpus*

Nous avons évalué notre approche sur deux corpus différents[2]. Le premier corpus regroupe les annotations RDF d'extraits de trois sources de données bibliographiques (DBLP, HAL et serveur interne de l'INRIA). Les annotations varient suivant la structuration des sources. Par exemple, certains nœuds contiennent les auteurs groupés et sont annotés par *SetOfPerson* alors que d'autres nœuds contiennt uniquement un auteur (annotés par *Person*). Les noms de conférences sont souvent localisés dans un seul nœud avec la date et le lieu de la conférence en question (nœuds annotés par *PartOf-Speech* et indexés par *Conference*, *Date* et *Location*), etc. L'ensemble des annotations constitue près de 10.000 triplets RDF.

2. http ://wwwdi.supelec.fr/~ bennacer/SHIRI/datasets.html

Nous avons soumis un ensemble de 5 requêtes pour rechercher des conférences et éventuellement leurs dates, lieux, articles et auteurs correspondants. Le petit nombre de requêtes se justifie par rapport à la régularité de structuration des documents concernés et au nombre des concepts/relations avec lesquels ils ont été annotés. Le but étant ici d'étudier la capacité de l'approche à intégrer différentes sources de données au moment de l'interrogation. La figure 4(b) présente le rappel et la précision des réponses quand la distance d varie de 1 à 7. Les résultats montrent que pour $d = 3$, toutes les réponses sont atteintes avec 100% de précision. Mais si le seuil de distance est ≥ 4, la précision est de presque 0%. En effet, dans deux sources de données, chaque article se voit associé à toutes les conférences pour ce seuil.



(a) Rappel et précision en fonction du seuil d'ordre

(b) Rappel et précision en fonction de d

Figure 4. *Rappel et précision (corpus de références bibliographiques)*

La figure 4(a) présente le rappel et la précision obtenus pour $d = 4$ et un seuil d'ordre variant de 1 à 10, 10 étant l'ordre maximum atteint pour les requêtes soumises. À un seuil d'ordre i donné, les premiers (meilleurs) i ensembles de reformulations sont générés et exécutés par $DREQ$. Les résultats montrent que la précision diminue lorsque le seuil d'ordre augmente. Un rappel de 100% est atteint pour toutes les requêtes soumises après le 9^{me} seuil d'ordre pour 28 reformulations maximum par requête. Chacune des sources a une structure propre mais nous avons pu récupérer les réponses de toutes les sources grâce à l'adaptation des annotations et à la reformulation des requêtes utilisateurs.

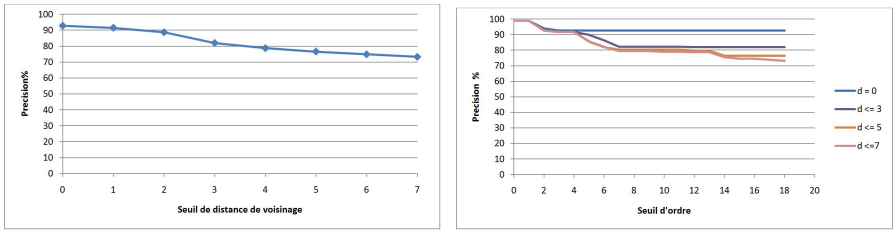
5.3. *Deuxième corpus*

Le deuxième corpus est constitué de 32 sites d'appels à communications annotés (environ 30.000 triplets RDF). Les annotations ont été générées par SHIRI Extract (Thiam *et al.,* 2009). Elles concernent des noms de conférences, des lieux, des dates, des membres de différents comités, ou des thèmes de recherche mais ne décrivent aucune relation sémantique entre les différentes instances. Ces annotations ont ensuite été automatiquement transformées par l'adaptateur de SHIRI-Querying pour se

conformer au modèle d'annotation. Enfin, nous avons soumis 15 requêtes de niveau domaine suivant l'ontologie $Call\text{-}For\text{-}Paper$ (Thiam *et al.,* 2008).

Aucune des requêtes de domaine soumises n'a de réponses directes dans la base d'annotations. Sans reformulation, le rappel est donc de 0%. Avec notre approche nous avons pu atteindre un rappel total de 56% pour une distance $d \leq 7$. Étant donné la grande hétérogénéité du corpus, ce rappel a été mesuré au pire cas (e.g. si la requête demandait le lieu d'un événement, nous avons considéré que tous les événements référencés dans la base avaient bien l'information lieu correspondante dans les documents).

La figure 5(a) décrit la précision moyenne des réponses quand la distance de voisinage d varie de 1 à 7. Sur ce corpus, la précision n'est pas en dessous de 72% pour une distance $d \leq 7$. En dépit du fait que les valeurs de seuil pertinentes pour d varient d'un corpus à un autre, notre expérimentation a bien validé l'hypothèse selon laquelle les relations sémantiques peuvent être retrouvés entre deux nœuds voisins en milieux très hétérogènes.



(a) Précision en fonction de d (b) Précision en fonction du seuil d'ordre

Figure 5. *Précision des réponses*

La figure 5(b) présente la précision moyenne des réponses pour les mêmes requêtes et pour plusieurs valeurs de seuil différentes pour d. La valeur d'ordre varie de 1 à 18, 18 étant l'ordre maximum atteint pour les requêtes soumises. Les résultats montrent que la précision diminue bien au fur et à mesure que le seuil de l'ordre augmente. Par exemple, la date de la conférence $CHES$ est correcte quand elle est localisée dans le nœud $PartOfSpeech$ qui contient la conférence et son nom, mais fausse quand elle est localisée dans un nœud $PartOfSpeech$ voisin. Les thèmes de la conférence sont aussi corrects s'ils sont localisés dans un nœud $SetOfTopics$ mais faux quand ils sont localisés dans un nœud $PartOfSpeech$ voisin, indexé par plusieurs concepts différents.

L'expérimentation montre que les deux reformulations élémentaires que nous proposons (par voisinage et par agrégation) se sont complétées l'une l'autre pour (1) contourner l'absence de relations sémantiques dans la base d'annotations et (2) remplacer les filtres sur les attributs des instances de domaine par une recherche de termes

dans les nœuds de document. La relation d'ordre proposée a aussi pu être validée comme critère pertinent pour trier les réponses.

Du point de vue de l'exécution des requêtes, nous employons une optimisation simple qui consiste à parcourir la base d'annotation avant toute interrogation afin d'identifier les metadonnées inexistantes. Cette étape réduit considérablement le nombre de reformulations (e.g. il n'y a pas de concept $SetOfConferences$ dans le corpus que nous avons utilisé). Sur les requêtes soumises, le temps de reformulation et d'exécution moyen est de 483ms sans optimisation et 266ms avec optimisation, sur une machine avec un processeur Core 2 Duo T9300 et 4Gb de RAM.

6. Conclusion et perspectives

Dans ce papier, nous avons présenté l'approche *SHIRI-Querying*, conçue pour soutenir la recherche sémantique de l'information dans les documents semi-structurés hétérogènes. Dans cette approche, des requêtes utilisateur basées sur ontologie sont reformulées pour atteindre les nœuds de documents appropriés (balises XML ou HTML). Cette reformulation permet de remédier à l'imprécision de localisation des instances dans les documents. Elle permet aussi de retrouver des instances reliées par des relations sémantiques même si ces relations n'existent pas dans la base d'annotations de départ. Nous avons défini un ordre entre les différentes reformulations possibles d'une requête utilisateur donnée. Cette relation d'ordre privilégie les requêtes qui préservent le plus possible la sémantique de la requête utilisateur.
Toutes les reformulations sont générées d'une manière dynamique avec l'algorithme $DREQ$ qui a été implémenté et testé sur deux corpus réels. Les résultats expérimentaux obtenus sont prometteurs. Ils montrent que le rappel augmente et que la précision diminue raisonnablement au fur et à mesure de la construction et de l'exécution ordonnée des requêtes reformulées. Même si le nombre de reformulations construites est exponentiel par rapport aux nombre de variables de la requête utilisateur, $DREQ$ génère des ensembles de reformulations de même ordre en temps polynomial.
Nous envisageons d'expérimenter notre approche en utilisant les d'annotations sémantiques obtenues par d'autres types d'annotateurs. Une autre perspective intéressante serait d'étendre notre approche pour résoudre certaines co-références et améliorer ainsi la qualité des réponses en combinant des éléments de réponses provenant de différents documents.

7. Bibliographie

Anyanwu K., Maduko A., Sheth A., « SemRank : Ranking Complex Relationship Serach Results on the Semantic Web », *International World Wide Web Conference (WWW)*, 2006.

Bhagdev R., Chapman S., Ciravegna F., Lanfranchi V., Petrelli D., « Hybrid Serach : Effectively Combining Keywords and Semantic Searches », *European Semantic Web Conference*, 2008.

Castells P., Fernàndez M., Vallet D., « An Adaptation of the Vector-Space Model for Ontology-Based Information Retreival », *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, n° 2, p. 20-27, 2007.

Cimiano P., Ladwig G., Staab S., « Gimme'The Context : Context Driven Automatic Semantic Annotation With C-PANKOW », *WWW conference*, 2005.

Corby O., Dieng-Kuntz R., Gandon F., Faron-Zucker C., « Searching the semantic web : Approximate query processing based on ontologies », *IEEE intelligent systems journal*, vol. 21, n° 1, p. 20-27, 2006.

Etzioni O., Cafarella M. J., Downey D., Popescu A.-M., Shaked T., Soderland S., Weld D. S., Yates A., « Unsupervised named-entity extraction from the web : An experimental study », *Artificial Intelligence*, vol. 165, n° 1, p. 91-134, 2005.

Hristidis V., Koudas N., Papakonstantinou Y., D. S., « Keyword proximity search in XML trees », *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, n° 4, p. 525-539, 2006.

Hurtado C.-A., Poulovassilis A., Wood P.-T., « A Relaxed Approach to RDF Querying », *International Semantic Web Conference, ISWC*, 2006.

Lamberti F., Sanna A., Demartini C., « A Relation-Based Page Rank Algorithm for Semantic Web Search Engines », *IEEE Transactions on Knowledge and Data Engineering*, 2009.

N appil a T., J arvelin K., Niemi T., « A tool for Data Cube Construction from Structurally Heterogeneous XML Documents », *Journal of the American Society for Information Science and Technology*, 2007.

Popov B., Kiryakov A., Kirilov A., Manov D., Ognyanoff D., Goranov M., « KIM - Semantic Annotation Platform », *Journal of Natural Language Engineering*, vol. 10, n° 3, p. 375-392, 2004.

Rocha C., Schwabe D., ao M. P. A., « A Hybrid Approach for Searching in the Semantic Web », *International World Wide Web Conference (WWW)*, 2004.

Thiam M., Bennacer N., Pernelle N., Lo M., « Incremental Ontology-Based Extraction and Alignment in Semi-Structured Documents », *DEXA*, p. 611-618, 2009.

Thiam M., Pernelle N., Bennacer N., « Contextual and Metadata-based Approach for the Semantic Annotation of Heterogeneous Documents », *ESWC-SeMMA workshop*, 2008.

W3C, « http ://www.w3.org », n.d.