Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

Centre Génie Industriel et Informatique (G2I)

UTILISATION DE LA LANGUE NATURELLE POUR L'INTERROGATION DE DOCUMENTS STRUCTURES

Xavier TANNIER, J. Jacques GIRARDOT, Mihaela MATHIEU

Décembre 2004

RAPPORT DE RECHERCHE
2004-400-010

MINISTÈRE DE L'ÉCONOMIE
DES FINANCES ET DE L'INDUSTRIE

Les rapports de recherche
du Centre G2I de l'ENSM-SE
sont disponibles en format PDF
sur le site Web de l'Ecole

G2I research reports
are available in PDF format
on the site Web of ENSM-SE

www.emse.fr

Centre G2I
Génie Industriel et Informatique

Division for
Industrial Engineering and Computer Sciences
(G2I)

Par courier :

By mail:

Ecole Nationale Supérieure des Mines de Saint-Etienne
Centre G2I
158, Cours Fauriel
42023 SAINT-ETIENNE CEDEX 2
France

Utilisation de la langue naturelle pour l'interrogation de documents structurés

Xavier Tannier, Jean-Jacques Girardot et Mihaela Mathieu
École Nationale Supérieure des Mines
158 Cours Fauriel, F-42023 Saint-Etienne, France
tannier,girardot,mathieu@emse.fr

Résumé :
Le langage de requête est l'indispensable interface entre l'utilisateur et l'outil de recherche. Simplifié au maximum dans les cas où les moteurs indexent essentiellement des documents plats, il devient fort complexe lorsqu'il s'adresse à des documents structurés et qu'il s'agit de définir des contraintes portant à la fois sur la structure et le contenu. L'approche ici-décrite propose d'utiliser la langue naturelle comme interface pour exprimer de telles requêtes.

L'article décrit dans un premier temps les différentes phases qui permettent de transformer (dans un cadre de recherche d'information) la requête en langage naturel en une représentation sémantique indépendante du contexte. Des règles de simplification adaptées à la structure et au domaine du corpus sont ensuite appliquées, permettant d'obtenir une forme finale, adaptée à une conversion vers un langage de requête formel. L'article décrit enfin les expérimentations effectuées et tire les premières conclusions sur divers aspects de cette approche.

Mots-clés : Documents structurés, XML, traitement du langage naturel, recherche d'information.

Abstract:
A query language is a necessary interface between the user and the search engine. Extremely simplified in the case of a retrieval performed on flat documents, this language becomes more complex when dealing with structured documents. Indeed we need then to specify constraints on both content and structure. In our approach we propose to use natural language as an interface to express such requests.

This paper describes first the different steps that we perform in order to transform (in an information retrieval framework) the natural language request into a context-free semantic representation. Some structure- and domain-specific rules are then applied, in order to obtain a final form, adapted to a conversion into a formal query language. Finally we describe our first experimentations and discuss different aspects of our approach.

Keywords: Structured documents, XML, natural language processing, information retrieval.

1 Introduction

Les outils de recherche d'information disponibles sur le Web mettent potentiellement à la disposition de n'importe quel individu du globe une très large portion de l'ensemble des documents accessibles au travers de l'Internet. Ces outils travaillent essentiellement en mode plat, c'est-à-dire qu'ils indexent soit des documents purement textuels, soit des documents structurés dans lesquels ils ne s'intéressent qu'au contenu, et non à la structure. Les langages de requête proposés aux utilisateurs ne permettent pas de faire référence à l'aspect structurel de ces documents : dans la quasi totalité des cas, une requête consiste en une liste de quelques mots-clés.

L'émergence récente d'un standard pour la représentation de données structurées, XML (eXtended Markup Language [22]), avec la possibilité des créer une infinité de structures au moyen des DTD (Document Type Definition) ou des Schémas XML, constitue une donne nouvelle dans ce paysage. Les documents XML, au travers d'étiquettes (balises), offrent aux informations une structure arborescente riche. XML permet aussi bien de traiter des données très structurées, à l'image de celles que l'on représente dans des bases de données relationnelles (et conçues pour être lues par des machines), que des textes beaucoup moins formellement structurés, tels que les articles scientifiques ou même les œuvres de la littérature romanesque. Dans le cas de la recherche d'information, les mots utilisés peuvent ainsi acquérir des sémantiques différentes en fonction des structures qui les contiennent.

Pour prendre en compte ces différents usages, deux approches différentes ont été imaginées : une approche orientée *bases de données* d'une part, plus adaptée pour sélectionner des données à fournir à des applications (comme XQL [11], XML-QL, XML-GL [3], Quilt [4] et surtout XQuery [24]) ; une approche orientée *recherche d'information* d'autre part ([26, 6, 21, 13]). Des approches "hybrides" ont été envisagées [8, 12]. Dans tous les cas, une recherche concernant la structure ne s'affranchit pas d'un langage de requête. La complexité inhérente de ce langage ainsi que la nécessité de connaître parfaitement la structure du document (sa DTD) le place de fait hors de portée de la quasi-totalité des utilisateurs potentiels.

Dans le cadre d'un projet de recherche, dont la finalité consiste à mettre à la disposition de chercheurs en cognitique, au travers du Web, un corpus de transcriptions de Français parlé en interaction (conversations orales), il nous est apparu que les besoins d'interrogation des chercheurs étaient clairs, et pouvaient s'exprimer sans ambiguïté dans la conversation courante. En revanche, la "traduction" de ces requêtes en un langage structuré est très lourde, voire impossible. De plus, il n'est guère envisageable de demander à ces chercheurs d'acquérir la maîtrise de langages de requête complexes tels que XQuery.

L'approche choisie consiste à permettre l'utilisation de la langue naturelle, quelque peu restreinte, pour l'expression des requêtes. Cette décision repose sur une analyse des spécificités des requêtes, qui font référence à des aspects structurels des documents, et à des phénomènes (ou combinaisons de phénomènes) que les chercheurs désirent retrouver. Cette combinaison de contraintes sur les contenus et la structure des documents est une des caractéristiques fondamentales qui permettent une mise en œuvre réaliste du processus proposé.

2 Langue naturelle et recherche d'information

L'idée de l'application des techniques de traitement automatique de la langue naturelle (TAL) au domaine de la recherche d'information n'est pas nouvelle (voir par exemple [15, 5, 16, 1, 9, 17]). Elle est été particulièrement étudiée dans le cadre des documents textuels, l'idée étant qu'une "compréhension" de la sémantique de la requête et de celles des textes à interroger devrait apporter des améliorations significatives dans ce cadre. Bien que les résultats obtenus n'aient pas été jusqu'alors à la hauteur des espérances [19, 17], il nous semble légitime d'envisager l'utilisation de la langue naturelle (LN) pour l'expression de requêtes portant sur des documents structurés. En effet :

– Il est possible, dans le cas d'une requête en LN portant à la fois sur la structure et le contenu, de séparer ces deux types de contraintes, et de les aborder au moyen des techniques les plus

appropriées ; dans le cas spécifique de l'application évoquée antérieurement, il n'y a pas vraiment de travail de compréhension de la sémantique du document à effectuer, mais juste une recherche de termes, dans la mesure où des précisions fortes sur certains aspects de la sémantique sont apportées par la structure.

– les bénéfices attendus sont bien plus significatifs que pour la RI traditionnelle, si l'on estime que n'importe quel utilisateur pourrait dès lors poser une requête relativement précise au moyen du langage naturel, requête qu'il ne saurait exprimer dans un langage formel structuré, dans la mesure où, en général, il ignore à la fois ce langage et les DTD des documents interrogés.

3 Description de notre approche

Notre but est de générer une requête en langage formel à partir d'une requête descriptive écrite en langue naturelle. Pour réaliser l'analyse de telles requêtes les étapes suivantes ont été réalisées :

– une analyse morpho-syntaxique (3.1) ;

– une analyse syntaxique et sémantique de la requête (3.2) ;

– une application des règles spécifiques (3.3) :

  ◇ une reconnaissance de quelques constructions typiques de requête (*e.g.: Rechercher + objet*) ou du corpus (*e.g.: "un article écrit par [...]"* fait référence à une balise *auteur* si elle existe) ;

  ◇ une distinction entre les éléments sémantiques qui seront projetés sur la structure et, respectivement, sur le contenu ;

– un traitement des relations existant entre différents éléments reconnus à l'étape précédente (3.4) ;

– la construction de la requête en langage formel d'interrogation (3.5).

3.1 Analyse morpho-syntaxique

L'analyse morpho-syntaxique est le processus visant à marquer les mots d'un texte avec leur catégorie grammaticale (nom, verbe, adjectif...), connaissance indispensable à l'analyse linguistique du texte. Nous avons utilisé l'outil TreeTagger [14] qui effectue cette tâche pour de nombreuses langues dont le Français. Par exemple, pour la requête suivante :

(1) *Trouver les titres d'articles qui parlent de sémantique.*

... la sortie de TreeTagger est donnée par la figure 1.

3.2 Analyses syntaxique et sémantique

L'analyse syntaxique s'opère par la mise en place d'un ensemble de règles décrivant les constructions grammaticales les plus courantes dans les requêtes et les questions. Nous avons choisi d'utiliser des règles *hors-contexte* : elles définissent quelle séquence d'éléments (sur la partie droite) est nécessaire pour obtenir un seul nouvel élément (sur la partie gauche). La figure 2 liste à titre d'exemple les règles qui sont déclenchées lors de l'analyse de la requête (1).

Une application récursive de ces règles aboutit à un *arbre syntaxique*, représenté à la figure 3. Notons qu'avec cet ensemble de règles deux constructions différentes sont possibles : la proposition relative peut être attachée au nom *"article"* (comme indiqué dans la figure) mais aussi au nom *"titre"*. En pratique les deux arbres sont explorés.

Trouver	VERBE(INF)	trouver
les	DET	le
titres	NOM	titres
d'	PREP	de
articles	NOM	article
qui	PRO:REL	qui
parlent	VERBE(PRES)	parler
de	PREP	de
sémantique	NOM	sémantique

FIG. 1: analyse morpho-syntaxique de la phrase (1) par TreeTagger. *Trouver* est un verbe à l'infinitif, *de* est une préposition et *qui* un pronom relatif.

$GN \rightarrow DET?\ NOM$
$GN \rightarrow GN\ PREP\ GN$
$GN \rightarrow GN\ PROP\_REL$
$PROP\_REL \rightarrow PRO{:}REL\ GV$
$GV \rightarrow VERBE\ PREP?\ GN$
$P \rightarrow VERBE(IMP)\ GN$

FIG. 2: Exemple de règles hors contexte, avec P = phrase, GN = groupe nominal, $PROP\_REL$ = proposition relative, GV = groupe verbal. Le point d'interrogation "?" signifie que l'élément est optionnel.

Cette opération nous donne une structure syntaxique, mais nous avons besoin d'un peu de sémantique pour avoir une idée des relations qui existent entre les mots. Dans ce but, nous utilisons une implémentation très simple de la Théorie de Représentation du Discours (DRT) [10] (pour une description plus accessible voir [2], volume II). Dans la DRT la représentation sémantique du discours est décrite par une "boîte" à deux étages appelée "Structure de Représentation de Discours" (DRS). L'étage supérieur donne les référents, qui sont les éléments introduits par le discours ; le niveau inférieur représente les conditions concernant les référents.

La figure 4 montre un exemple typique de DRS. Dans cet exemple, les termes *"Napoléon"*, *"Austerlitz"*, *"bataille"* et le verbe *"gagner"* (événement e) sont des référents du discours, représentés par des lettres à l'étage supérieur et décrits par des prédicats logiques au-dessous. Les autres conditions donnent enfin l'agent et l'objet de l'événement (respectivement *"Napoléon"* et *"bataille"* pour l'événement *"gagner"*) ainsi que le lieu de la bataille.

Pour générer automatiquement une DRS représentant une phrase complète, nous attribuons une DRS de base à chaque mot, en fonction de sa catégorie grammaticale. La figure 5 reprend notre exemple de la phrase (1) et donne trois DRS unitaires pour un nom, un verbe et une préposition.

Les règles syntaxiques sont ensuite enrichies par des actions sémantiques. Dans notre exemple, la règle suivante :

$$GV \rightarrow VERB\ PREP?\ GN,$$

qui s'applique à la DRS de la figure 5 est associée aux identités suivantes : $e_1 = e_2$ et $x = y$, ce qui produit l'arbre sémantique décrit à la figure 6.

Nous manquons de place pour fournir l'arbre sémantique complet obtenu pour cet exemple. La figure 7 donne la DRS finale. A noter qu'un référent a été ajouté, car il est implicite dans la phrase : l'*interlocuteur* qui donne un ordre quand on utilise un verbe au mode impératif ou infinitif (ici, *"trouver ..."*).

N.B. : L'ensemble de règles syntaxiques et sémantiques que nous utilisons comporte quelque 50 règles et évidement nous n'allons pas l'expliciter ici entièrement. L'accent a été mis sur les groupes

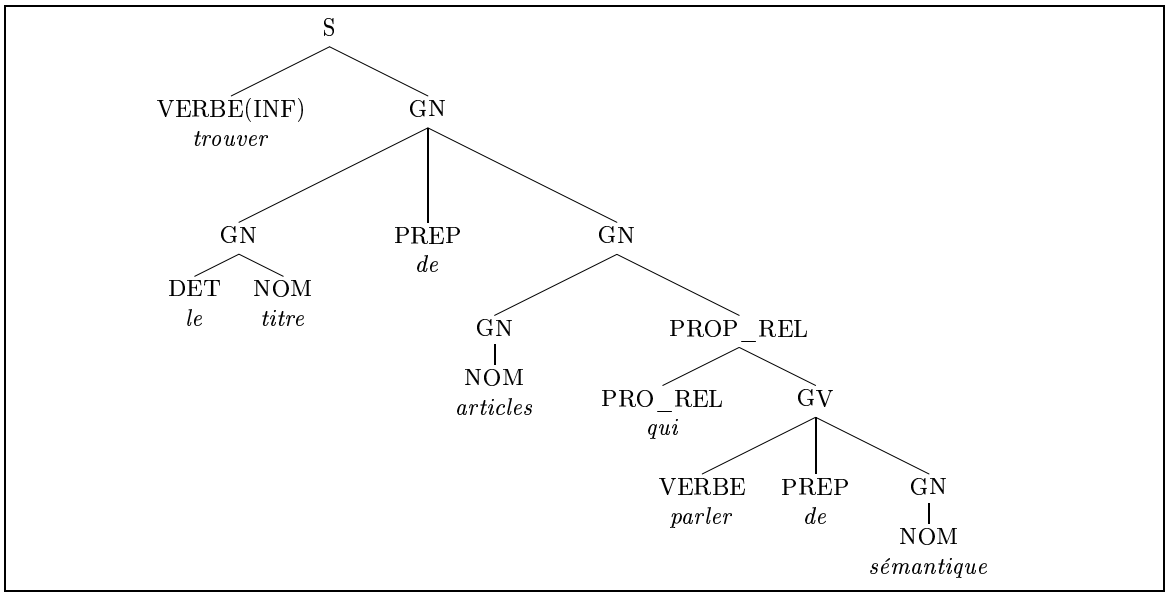


FIG. 3: Arbre syntaxique de la phrase (1), obtenu avec les règles de la figure 2.

e x y z
Napoléon(x)
bataille(y)
Austerlitz(z)
evt(e, gagner)
agent(e, x)
objet(e, y)
lieu(y, z)

FIG. 4: Représentation sémantique en DRT de la phrase : *"Napoléon gagne la bataille d'Austerlitz"*

nominaux qui sont souvent beaucoup plus riches en sens que les groupes verbaux (au moins en termes de recherche d'information). Les propositions relatives, les constructions prépositionnelles sont aussi très importantes parce qu'elles marquent la structure de la requête, ce que nous ne voulons pas perdre. Pour les requêtes complexes une analyse complète est souvent impossible. Dans ce cas, seuls les groupes nominaux sont analysés et les verbes sont ignorés.

La DRS que nous obtenons à cette étape ne peut pas encore être utilisée pour une requête formelle. Quelques règles spécifiques à la recherche d'information doivent être appliquées.

3.3 Règles spécifiques

La construction sémantique peut être réduite en prenant en compte quelques cas particuliers, parmi lesquels :

1. Les **verbes de requête** tels que *"vouloir"*, *"trouver"*... Un dictionnaire décrivant la sémantique entre ces verbes et les termes qui leur sont associés nous permet de reconnaître l'élément concerné, qui devra être sélectionné à la fin du processus comme bonne réponse à la requête :

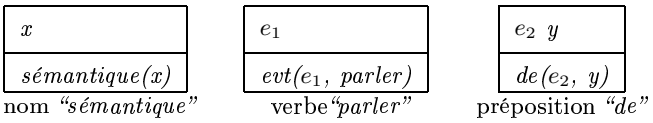


FIG. 5: exemples de DRS unitaires

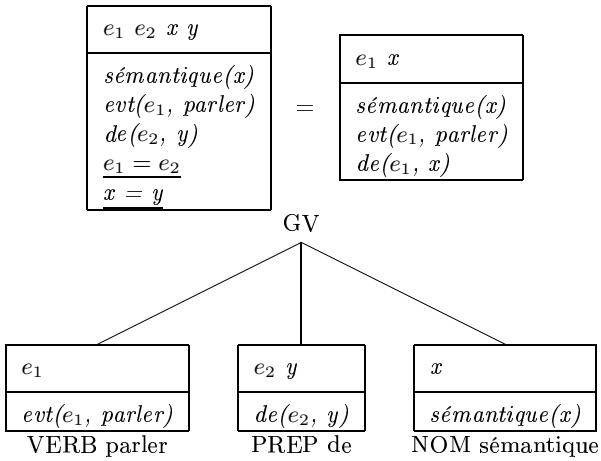
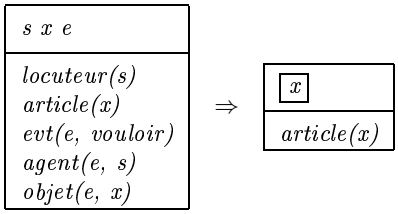


FIG. 6: Etapes de la construction de DRS pour le groupe verbal "*parler de sémantique*"

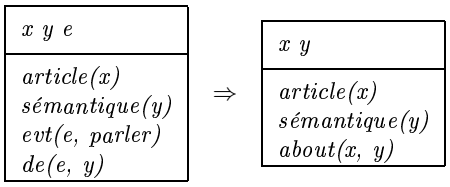
(2) Je **veux** un article.



Ici nous savons que le verbe *"vouloir"* signifie que son objet (*"article"*) doit être séléctionné et cette nouvelle information est représentée par un référent encadré. Le verbe lui-même ainsi que son agent (ici le *locuteur*, ou *"je"*) sont éliminés.

2. Les **verbes descriptifs** tels que *"parler de"*, *"concerner"*... Un autre dictionnaire contient l'information qui nous permet d'ajouter une nouvelle relation, que nous nommons *about* :

   (3) un article qui **parle de** sémantique.



Le verbe est également supprimé dans ce cas.

$x\ y\ z\ s\ e_1\ e_2$
$evt(e_1,\ trouver)$
$evt(e_2,\ parler)$
$titre(x)$
$article(y)$
$sémantique(z)$
$interlocuteur(s)$
$agent(e_1,\ s)$
$objet(e_1,\ x)$
$de(x,\ y)$
$agent(e_2,\ y)$
$de(e_2,\ z)$

FIG. 7: DRS pour la phrase (1)

3. Les **verbes de relation topologique** comme *"contenir"*, *"inclure"*... Si un tel verbe a un agent et un objet, une relation appropriée est construite entre ces deux éléments et le verbe est enlevé :

   (4) une section qui **contient** une figure...

$x\ y\ e$
$section(x)$
$figure(y)$
$evt(e,\ contenir)$
$agent(e,\ x)$
$objet(e,\ y)$
...

$\Rightarrow$

$x\ y$
$section(x)$
$figure(y)$
$contient(x,\ y)$
...

4. Quelques **règles sémantiques** spécifiques au corpus doivent être rajoutées afin de reconnaître certaines constructions linguistiques précises.

   (5) Un document **écrit par** Untel.

   En supposant qu'existe dans notre corpus une balise *auteur*, une règle *ad hoc* impose la transformation suivante :

$x\ y\ e$
$document(x)$
$Untel(y)$
$evt(e,\ écrire)$
$agent(e,\ y)$
$objet(e,\ x)$

$\Rightarrow$

$x\ y\ z$
$document(x)$
$Untel(y)$
$auteur(z)$
$contient(z,\ y)$
$contient(x,\ z)$

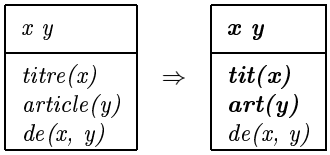
   N.B. : Dans cet exemple il ne s'agit plus de règles à caractère *linguistique*, mais de transformations nécessaires dans le cadre de la recherche d'information. Dans ce dernier exemple les deux prédicats *contient* n'ont pas une réelle signification linguistique, mais expriment une contrainte structurelle dans le document XML.

5. Les **mots entre guillemets** sont considérés comme une expression non-séparable et sont groupés ensemble en une seule variable.

(6) Nous chercherons la répétition “non non non” à proximité de “ah oui !”.

6. Et surtout, **un terme reconnu comme une étiquette de DTD (ou synonyme)** est marqué comme tel (ici par un prédicat en gras dans une DRS, avec *tit* signifiant *“titre”*).

   (7) le **titre** d'un **article**...



   Un dictionnaire de synonymes est utilisé. Ce dictionnaire n'est absolument pas destiné à une utilisation générique ; il est spécifique au corpus. En effet les noms des balises sont rarement de vrais mots, mais plutôt des abréviations (*ts* pour *titre de section*, *par* pour *paragraphe*, etc.).

La figure 8 montre une application de quelques-unes des règles spécifiques à notre exemple de DRS. Rappelons la requête initiale :

(1) *Trouver les titres d'articles qui parlent de sémantique.*

Nous supposons que notre dictionnaire nous indique que les mots *“titre”* et *“article”* désignent les balises *tit* and *art*.

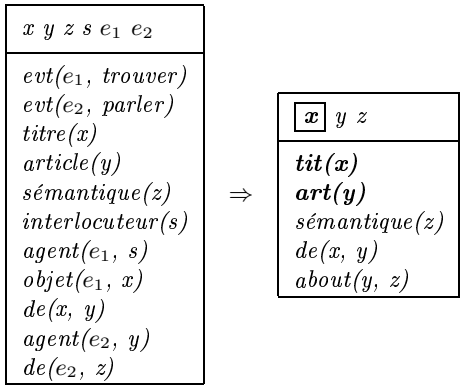


FIG. 8: DRS de la phrase (1) avant et après l'application des règles spécifiques. Le référent x est sélectionné (règle 1), l'article y traite de *“sémantique”* (relation *about*, règle 2) et les termes *“article”* et *“titre”* sont reconnus comme identificateurs de balises *art* et *tit* (règle 6).

Dans cette nouvelle DRS nous pouvons distinguer clairement un *nom de balise*, qui est attaché à la structure du document (en gras), et un *terme*, comme *“sémantique”*, qui est supposé faire partie du contexte textuel du document.

3.4 Analyse structurelle

A cette phase de l'analyse, il existe encore des relations binaires entre certains référents qui n'ont pas été traitées par une règle spécifique (exemple : la relation *de(x, y)* de la figure 8). Ces relations ont pour l'utilisateur des sémantiques particulières, que le système ne peut connaître.

Soit $R(x, y)$ une relation binaire entre les référents x et y. Afin de prendre en compte aussi bien les relations “connues” (par exemple les relations d'ordre temporel comme *“avant”*...) que les autres, nous appliquons les heuristiques suivantes :

1. x et y font référence à des balises XML (des éléments structurels) :

   (a) si la relation R est connue, aucune action n'est effectuée.

   e.g.: Un **paragraphe** après une **figure**:

x y
par(x)
fig(y)
après(x, y)

   (b) sinon, le fait qu'il existe une relation représente en soi une information ; la DTD nous permet de trouver de quelle relation il peut s'agir : ainsi dans notre exemple il est possible de savoir que l'élément *tit* (x) est contenu dans l'élément *art* (y).

   e.g.: Un **titre** d' **article**:

x y
tit(x)
art(y)
de(x, y)

   $\xrightarrow[analysis]{DTD}$

x y
tit(x)
art(y)
contains(y, x)

2. La relation existe entre une balise (disons x) et un terme (y) :

   (a) si R est connue, nous ajoutons une balise correspondant à un nom d'élément quelconque ('*'). Ce nouvel élément est lié à y par la relation *about*.

   e.g.: Un **paragraphe** avant "Attends" :

x y
par(x)
Attends(y)
avant(x, y)

   $\rightarrow$

x y z
par(x)
Attends(y)
****(z)***
about(z, y)
avant(x, z)

   Le paragraphe doit précéder dans le document XML l'élément contenant le mot "Attends".

   (b) si R n'est pas connue, on la conserve (voir les remarques pour la règle 3).

3. La relation R se rapporte à deux termes (*terme* est utilisé ici par opposition à *nom de balise* de la DTD). Aucun aspect structurel n'est impliqué, et aucun traitement particulier n'est prévu. Cependant, la relation peut être utile (sur le plan sémantique) au moteur de recherche, qui peut en faire usage ou non.
   e.g.: Une longue croisière *sur* le Danube :

x y z
croisière(x)
Danube(y)
long(z)
rel_ adjectif(x, z)
sur(x, y)

   Si le moteur de recherche est capable de gérer de telles relation (*rel_ adjectif* et *sur*), il est utile de savoir, par exemple, que la construction *"croisière sur le Danube"* est préférable aux mots *"croisière"* et *"Danube"* séparés.

Parmi ces règles, seule la règle 1b s'applique à notre exemple, et la DRS finale obtenue est donnée à la figure 9.

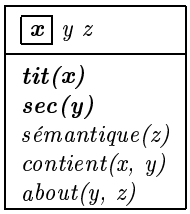


FIG. 9: DRS finale pour la phrase (1).

3.5 Requête formelle structurée

A la fin de l'étape linguistique, notre but est d'obtenir une requête qui se traduise aisément dans un langage de requête structuré. Nous avons emprunté aux langages existant le concept de séparation en clauses de type SELECT-FROM-WHERE (SQL) pour modéliser la requête. Celle-ci (exprimée en pratique en XML) s'exprime ainsi au travers des quatre fragments :

- la clause `from` fait référence aux étiquettes des éléments et exprime des indications sur leur chemin d'accès (expressions XPath[23]) ;
- la clause `select` exprime quels résultats doivent être fournis à l'utilisateur ; ces éléments doivent être également présents dans la clause `from` ;
- la clause `where` traduit les relations à satisfaire entre éléments et/ou variables, découvertes lors de l'analyse ;
- les identificateurs de la clause `variables` représentent les termes intervenant dans les autres clauses.

N.B. : La clause `select` indique seulement les résultats désirés ; les besoins éventuels de présentation ou de mise en page des résultats n'entrent pas dans le cadre de la recherche d'information et peuvent être satisfaits par des mécanismes externes au processus d'extraction de l'information.

La transformation de la DRS en une telle requête est relativement directe : les noms des balises XML ont déjà été repérés (en gras dans les exemples), ainsi que les données sélectionnées (référents encadrés dans la DRS). Les variables correspondent aux prédicats unaires qui ne représentent pas des noms de balises, et l'on retrouve dans la clause `where` l'ensemble des autres conditions.

Le système génère ainsi une requête, représentée en XML. Comme on peut s'y attendre, cette requête est peu lisible, et nous la traduisons en une forme telle que celle de la figure 10 (obtenue par exemple au travers d'une feuille de style XSLT [25]).

```
FROM y = /art,
     x = y//tit,
WHERE about(y, z)
VARIABLES z = "sémantique"
SELECT x
```

FIG. 10: Interrogation en langage formel, traduite de la DRS de la figure 9 (requête (1)). `x` et `y` sont des nœuds de types *art* et *tit* respectivement, et `z` est une variable représentant le terme *"sémantique"*.

4 Le processus de recherche d'information

Nous disposons à ce stade d'une requête susceptible d'être transformée en une syntaxe à définir et soumise à un moteur de recherche. Il nous faut cependant garder en mémoire le fait que la recherche effective de l'information reste à effectuer, et qu'un langage orienté "base de données" tel que XQuery peut ne pas suffire :

- Des relations, telles que *about* générée dans notre exemple, doivent être implémentées d'une manière ou d'une autre.
- Il serait utile (voire indispensable) que le moteur de recherche "comprenne" les contraintes linguistiques générées par l'analyse (cf. section 3.4).
- La requête étant traduite de la langue naturelle, les chemins XPath de la clause `from` ne sont pas nécessairement à prendre au sens strict ; ainsi, dans une requête telle que *"un paragraphe qui parle de sémantique"*, d'autres éléments, telles qu'une section ou une figure, peuvent convenir, voire même s'avérer plus pertinents.

5 Commentaires

5.1 Expérimentations

Dans un but de validation, dans la mesure où le *langage intermédiaire* n'est pas entièrement défini et stabilisé, l'outil a été interfacé avec un moteur de recherche *ad hoc*, minimaliste, qui indexe les mots et les positions des éléments des documents XML. Cette expérience a permis de valider l'approche sur un premier corpus, proposé à INEX 2004 [7] dans la catégorie "interrogation en langue naturelle". Les résultats [20] se sont avérés encourageants.

D'autres tests s'effectuent à l'heure actuelle sur le corpus mentionné dans l'introduction (ACI-TTT Corpus). Ce corpus n'étant que partiellement disponible (petit nombre de documents, DTD non stabilisée), la validation s'effectue sur un petit nombre de requêtes prototypiques, fournies par nos partenaires ; elle consiste à comparer le code fourni par le traducteur avec celui qui aurait été écrit "à la main" par un utilisateur connaissant bien le corpus, la structure de documents et le langage de requête. Là encore, les résultats se montrent positifs, le langage intermédiaire fourni semblant relativement proche de l'optimum envisageable pour le moteur en cours d'écriture.

5.2 Connaissances extra-linguistiques

Durant les étapes décrites dans les sections précédentes, nous avons utilisé différentes sortes d'informations concernant le corpus exploré. Cette connaissance doit être modélisée pour chaque nouvel ensemble de documents[1], et par conséquent empêche une application rapide et facile de la technique à tout type de documents XML ; pour cette raison nous avons voulu réduire autant que possible cette nécessité.
Malgré cela, nous pensons que les points suivants représentent la connaissance minimale qu'il est nécessaire de posséder pour effectuer une analyse appropriée des requêtes :

- La DTD. La connaissance de cette définition de la structure est la condition *sine qua non* du fonctionnement de notre méthode.
- Dans le cas (fréquent) où les noms de balises ne sont pas de "vrais" mots (voir section 3.3), un dictionnaire fournissant la sémantique de la DTD, car nous considérons que l'utilisateur ne connaît pas la structure du corpus.
- Eventuellement un dictionnaire de synonymes acceptables pour les noms de balises (*ex :* papier = article = document, etc.).

[1] Nous appelons un *nouvel* ensemble de documents un corpus avec une DTD différente et un domaine différent.

- Certaines locutions sémantiques (*ex :* “une liste de mots-clés” = “mots-clés”), ceci dans le but d'éviter le bruit provoqué par une génération erronée de termes (ici, *liste* n'est ni une balise ni un terme à rechercher dans le texte).

- Des structures ontologiques très simples (*ex :* “un roman écrit par Marcel Proust” = “un roman dont l'auteur est Marcel Proust” – en supposant que *roman* et *auteur* représentent des balises.

Il semble évident que mieux les informations concernant le corpus sont modélisées, meilleure pourra être l'analyse. En ce sens, une ontologie du domaine traité pourrait être utile, mais les points énumérés représentent des connaissances qu'il semble difficile de contourner.

5.3 Limites

5.3.1 Langage Naturel

Pour la méthode présentée l'expression "langue naturelle", avec ce qu'elle implique en termes d'expressivité et de généralité d'utilisation, est très certainement exagérée ; il convient de préciser les limites du procédé :

- le mécanisme marche "bien" pour des questions précises et bien structurées, d'autant mieux, d'ailleurs, que la requête est courte et fait référence aussi bien à des aspects de la structuration qu'au contenu du document ; en dehors de ces restrictions on s'expose à la génération de bruit ou de silence.

- le système doit être adapté à chaque cas particulier : trois domaines doivent être définis et qualifiés, et la qualité de cette définition influe sur les performances de l'outil ; ce sont :

  - ◇ le domaine du document
  - ◇ le domaine de l'utilisateur, qui peut également employer un vocabulaire propre (ex : "réplique" ou "tour de parole" selon son domaine de recherche)
  - ◇ le recouvrement entre le langage naturel et la structure du document (qu'est-ce qu'un "titre", que le "corps du document", que veut dire "dans le texte", etc.).

  Ces connaissances, comme cela a été signalé, sont pour l'instant codées sous forme de règles ou de contraintes programmatiques au sein des logiciels réalisés.

5.3.2 Bruit et silence

Nous avons effectué trop peu d'expériences pour pouvoir relier quantitativement bruit et silence à des caractéristiques précises de la nature des corpus et de la forme des requêtes. Cependant, il est clair que l'efficacité et les performances de la méthode décrite leur sont fortement liées. Expérimentalement, nous constatons que bruit et silence peuvent augmenter pour des requêtes longues ; plusieurs phénomènes interviennent :

- Dans une phrase longue, la syntaxe linguistique utilisée s'éloigne des constructions spécifiques à une requête de recherche d'information et s'approche de phrases du langage commun, avec des anaphores, des insinuations pragmatiques, dont les finesses sont hors de portée d'une analyse computationnelle. Cela conduit généralement a un bruitage fort des résultats. L'exemple suivant est inspiré d'une requête de la collection INEX 2004 :

  (8) Nous écrivons un rapport sur les méthodes de réduction de dimensions en recherche d'information. Des exemples de ces méthodes sont le LSI (latent semantic indexing) qui améliore le rappel, ou la projection aléatoire qui ne modifie pas trop les distances. La filtration ne nous intéresse pas.

– Dans une requête très structurée l'analyse syntaxique aboutit à plusieurs interprétations ambigües.

(9) [une interruption$_{N_1}$ puis le mot "Attends"$_{N_2}$]$_{GN}$ [dans des discussions$_{N_3}$ de moins de 10 phrases$_{N_4}$]$_{GP}$, où interviennent 3 personnes$_{REL}$.

Dans cet exemple pourtant simple, quatre analyses syntaxiques correctes sont possibles : Le groupe prépositionnel GP peut être attaché à N_2 ou à l'ensemble GN. De plus la proposition relative REL peut être liée à N_3 ou à N_4. Deux ambiguïtés produisent 2^2 arbres syntaxiques. Pour limiter ce problème, nous offrons à l'utilisateur la possibilité de parenthéser les expressions pour empêcher certains regroupements indésirables (par exemple des parenthèses autour du GN évitent l'attachement du GP à N_2).

– Enfin la multiplication des relations non interprétées ne permet pas de prendre en compte toutes les informations fournies par l'utilisateur.

6 Conclusion

Cet article a présenté une méthode de transformation de la requête d'un utilisateur, formulée en langage naturel, dans le cadre de la recherche d'information dans des documents structurés comme XML.

Nos premières expérimentations, en dépit des limites évoquées, tendent à valider l'approche. La continuation de nos travaux passe par deux tâches :

– développer l'interface avec un moteur de recherche bien adapté à nos objectifs.

– développer un formaliste pour la représentation des domaines spécifiques, liés à un domaine de référence, un type d'utilisateur, ou une représentation des documents.

Le système ainsi complété devrait permettre de passer à un stade plus avancé d'expérimentations, fournissant des résultats permettant une première évaluation qualitative et quantitative de cette approche.

References

[1] Avi Arampatzis, Th.P. van der Weide, C.H.A. Koster, and P. van Bommel. Linguistically-motivated Information Retrieval. In Allen Kent, editor, *Encyclopedia of Library and Information Science*, volume 69, pages 201–222. Marcel Dekker, Inc., New York, Basel, December 2000.

[2] Patrick Blackburn and Johan Bos. *Representation and Inference for Natural Language; A first Course in Computational Semantics*. ComSem, 1999.

[3] Stefano Ceri, Sara Comai, Ernesto Damiani, Piero Fraternali, Stefano Paraboschi, and Letizia Tanca. XML-GL: a Graphical Language for Querying and Restructuring XML Documents. In *Proceedings of the 8th International WWW Conference, WWW8*, Toronto, Canada, May 1999. International World Wide Web Conference Committee (IW3C2).

[4] Don Chamberlin, Jonathan Robie, and Daniela Florescu. Quilt: An XML Query Language for Heterogeneous Data Sources. In *Proceedings of WebDB 2000 Conference*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[5] Susan Feldman. NLP Meets the Jabberwocky: Natural Language Processing in Information Retrieval. Online, May 1999. http://www.onlinemag.net/OL1999/feldman5.html.

[6] Norbert Fuhr and Kai Großjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In W.B. Croft, D. Harper, D.H. Kraft, and J. Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 172–180, New York City, New York, USA, 2001. ACM Press, New York City, NY, USA.

[7] Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltàn Szlàvik, editors. *The Third Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, Schloss Dagstuhl, International Conference And Research Center For Computer Science, Germany, December 2004.

[8] Torsten Grabs and Hans-Jörg Schek. ETH Zürich at INEX: Flexible Information Retrieval from XML with PowerDB-XML. In N. Fuhr, N. Gövert, G. Kazai, and M.Lalmas, editors, *Proceedings of the First Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, Schloss Dagstuhl, International Conference And Research Center For Computer Science, Germany, December 2002.

[9] Christian Jacquemin and Pierre Zweigenbaum. Traitement automatique des langues pour l'accès au contenu des documents. In Toulouse Cepadues, editor, *Le document en sciences du traitement de l'information*, chapter 4, pages 71–109. Jacques Le Maître, Jean Charles and Catherine Garbay, 2000.

[10] Hans Kamp and Uwe Reyle. *From discourse to logic*. Kluwer Academic Publisher, 1993.

[11] Jonathan Robie, Joe Lapp, and David Schach. XML Query Language (XQL). http://www.w3.org/TandS/QL/QL98/pp/xql.html.

[12] Karen Sauvagnat. XFIRM : Un Modèle Flexible de Recherche d'Information pour le stockage et l'interrogation de documents XML. In *Actes de la 1ère Conférence en Recherche d'Information et Applications, CORIA'04*, pages 121–142, Toulouse, France, March 2004. IRIT, Toulouse.

[13] Torsten Schlieder and Holger Meuss. Querying and Ranking XML Documents. *Journal of American Society for Information Science and Tecnology (JASIST), Special Topic Issue on XML and Information Retrieval*, 53(6):489–503, April 2002.

[14] Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, September 1994.

[15] Alan F. Smeaton. Information Retrieval: Still Butting Heads with Natural Language Processing? In M.T. Pazienza, editor, *Information Extraction – A Multidisciplinary Approach to an Emerging Information Technology*, volume 1299 of *Lecture Notes in Computer Science*, pages 115–138. Springer-Verlag, 1997.

[16] Alan F. Smeaton. Using NLP or NLP Resources for Information Retrieval Tasks. In Strzalkowski [18], pages 99–111.

[17] Karen Sparck Jones. What is the role of NLP in text retrieval? In Strzalkowski [18], pages 1–24.

[18] Tomek Strzalkowski, editor. *Natural Language Information Retrieval*. Kluwer Academic Publisher, Dordrecht, NL, 1999.

[19] Tomek Strzalkowski, Fang Lin, Jin Wang, and Jose Perz-Carballo. Evaluating Natural Language Processing Techniques in Information Retrieval. In Strzalkowski [18], pages 113–145.

[20] Xavier Tannier, Jean-Jacques Girardot, and Mihaela Mathieu. Analysing Natural Language Queries at INEX 2004. In Fuhr et al. [7], pages 196–203.

[21] Anja Theobald and Gerhard Weikum. The Index-based XXL Search Engine for Querying XML Data with Relevance Ranking. In *Proceedings of the 8th International Conference on Extending Database Technology (EDBT)*, pages 477–495, Prague, Czech Republic, March 2002.

[22] Extensible Markup Language (XML). World Wide Web Consortium (W3C) Recommandation. http://www.w3.org/TR/REC-xml/.

[23] XML Path Language (XPath). World Wide Web Consortium (W3C) Recommandation. http://www.w3.org/TR/xpath.

[24] XQuery 1.0: An XML Query Language. World Wide Web Consortium (W3C) Working Draft. http://www.w3.org/TR/xquery.

[25] XSL Transformation (XSL). World Wide Web Consortium (W3C) Recommandation. http://www.w3.org/TR/xslt/.

[26] Jeans E. Wolff, Holger Florke, and Armin B. Cremers. Searching and Browsing Collections of Structural Information. In *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries (ADL)*, pages 141–150, Washington, D.C., USA, May 2000. IEEE Computer Society.

Ecole Nationale Supérieure des Mines de Saint-Etienne
Centre G2I
158, Cours Fauriel
42023 SAINT-ETIENNE CEDEX 2

www.emse.fr