Prédicats logiques / prédicats linguistiques pour la consultation de base de données en langue naturelle

Béatrice BOUCHOU, Denis MAUREL, Barbara KALTZ

Abstract. In this paper, we are suggesting to tackle the much debated issue of processing natural language queries of data bases in a different perspective. After discussing TEAM, a system typical of an analysis based on logical predicates, we present an application we are currently developing. Our approach is then illustrated through three examples of queries, previously used and discussed in TEAM. The system we are proposing appears to be less complex and to generate quick and pertinent answers to queries.We are presently working on a more general prototype of the system. Our aim is not to create yet another ad hoc product. This is why we have chosen to extend our linguistic study to both French and German.

Keywords: database, queries, NLP, finite state automata, transducers.

Mots-clés : base de données, interrogations, Taln, automates à nombre fini d'états, transducteurs.

1. Introduction

L'accès à l'information stockée dans les bases de données est l'un des principaux axes grand public de l'informatique, c'est pourquoi le défi

✉ Béatrice BOUCHOU : e-mail : bouchou@univ-tours.fr
LI (Laboratoire d'Informatique de l'Université de Tour), E3i,
64, avenue Jean Portalis, 37200 Tours

Denis MAUREL : e-mail : maurel@univ-tours.fr
LI (Laboratoire d'Informatique de l'Université de Tour), E3i,
64 avenue Jean Portalis, 37200 Tours

Barbara KALTZ : e-mail : kaltz@univ-tours.fr
LTI (Langues et Technologie de l'Information), Université de Tours,
Rue des Tanneurs, 37000 Tours,

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

de la langue naturelle est important dans ce domaine. Le traitement automatique des questions en langue naturelle posées à une base de donnée est donc un sujet d'études déjà ancien. Beaucoup de propositions ont été développées dans ce domaine au cours des années 80, accompagnant la maturation des principes du calcul symbolique logique. Ces systèmes font d'abord une analyse syntaxique complète de la question, puis ils extraient de l'arbre syntaxique (et d'autres sources d'information) ce qui est nécessaire à l'interprétation de la question en termes d'éléments de la base de données.

Il est intéressant de noter cependant que les premiers travaux dans ce domaine utilisent un lexique, pour détecter des mots clefs dans la question, et la théorie syntaxique de Z.S. Harris, pour trouver une réponse (BASEBALL, cf. Green *et al.*, 1960). C'est ce que nous faisons, renouant ainsi avec une piste de travail abandonnée depuis. La limite de la technique mise en œuvre dans BASEBALL réside dans son lien trop étroit avec l'information traitée. En d'autres termes, on n'a obtenu de bons résultats de cette manière que pour des produits « *ad hoc* », conçus dans un but trop particulier (Sabah 1997). Nous répondons à ce problème en utilisant un dictionnaire électronique contenant le vocabulaire du domaine et une copie complète de la base (voir § 3.1). Ce qui est rendu possible par les avancées algorithmiques de la théorie des automates et transducteurs à nombre fini d'états. De cette façon, l'information n'est plus dans les codes de traitement, mais exclusivement dans un dictionnaire, en partie créé lors de l'installation du logiciel.

Les travaux qui suivirent BASEBALL s'attachèrent à exploiter automatiquement la base de données pour alimenter le lexique (Kaplan 1984, Grosz *et al.* 1987, Clifford 1988). Le système TEAM développé par B. Grosz et ses collègues a pour objectif principal cette « portabilité », c'est-à-dire la faculté de s'adapter aisément à chaque nouvelle base de données. Pour cela, il est conçu pour interagir avec deux types d'utilisateurs, un expert de la base de données, d'une part, l'utilisateur final, d'autre part. Le système est général et il construit sa connaissance de la base de données lors de son installation. Cela se fait par interaction avec l'expert de la base de données. L'essentiel de l'interprétation dépend des directives de cet expert. Nous adoptons la même démarche, mais en la simplifiant dans la mesure où nous réduisons la

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

participation de l'administrateur de la base au minimum : seule une bonne connaissance de sa base de données lui est demandée.

Le système TEAM est un exemple abouti de l'approche basée sur une analyse syntaxique complète, suivie d'analyses sémantiques sur les arbres obtenus. Toutes ces opérations sont réalisées dans un formalisme du type logique du premier ordre. Or, une fois isolés les constituants sémantiques pertinents dans la question, il apparaît que les trois quarts des nœuds de l'arbre syntaxique ne représentent que la « glue syntaxique » (Grosz *et al.* 1987, p. 203) de la phrase originelle, et n'apportent rien à l'interprétation. Et pourtant ces nœuds ont été pris en compte dans toute la première phase d'analyse. Un autre exemple axé sur la logique est décrit dans (Péquegnat *et al.* 1988) : cinq bases de connaissances y sont constituées, représentations logiques des données linguistiques, conceptuelles (la base de données) et sémantiques. Un système complet de réécriture est nécessaire pour chaque passage d'une représentation à une autre... Les traitements en sont alourdis d'autant. Nous ne passons pas par toutes ces étapes : nous exploitons une connaissance linguistique codée par les prédicats linguistiques, à l'aide de transducteurs.

À la fin des années 80, le tour de la question de l'interrogation de base de données en langue naturelle semblait avoir été fait et la recherche dans ce domaine s'est alors donné des buts plus généraux : le dialogue homme-machine, d'une part, l'approfondissement des outils d'analyse du langage naturel, d'autre part. Les avancées conjointes des outils linguistiques et informatiques permettent, aujourd'hui, d'aborder la question autrement, de façon plus performante et plus générale nous semble-t-il.

Dans la suite de cet article, nous présentons le système TEAM (§ 2) et le nôtre (§ 3 et § 4). Puis nous décrirons trois exemples d'interrogations plus élaborées, exemples repris du rapport sur TEAM (§ 5). Enfin, nous conclurons par des perspectives de recherche (§ 6).Comme nous ne souhaitons pas créer un nouveau produit *ad hoc*, nous avons décidé de mener parallèlement notre étude linguistique sur le français et l'allemand. Nous présenterons en notes quelques explications sur la partie allemande de notre travail.

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

2. Approches par prédicats logiques : le système TEAM

Le système TEAM est représentatif des propositions de consultation de base de données en langue naturelle formalisées à l'aide de la logique. Il représente pour nous une bonne base de comparaison dans la mesure où il est décrit en détails dans (Grosz 1987).

2.1. Les grandes lignes du système TEAM

Le système TEAM a été développé par une équipe de l'« Artificial Intelligence Center, SRI International, Menlo Park, CA 94025, USA ». Il est présenté comme le résultat de quatre années de travaux, effectués dans le but principal d'améliorer la « transportabilité » des systèmes d'interrogation de bases de données en langue naturelle.

Le principe essentiel adopté dans ce but est de concevoir un système général, dans un premier temps, et de confier à un spécialiste de la base de données le soin de compléter ce système en fonction de cette base, dans une phase d'installation.

2.2. Phase d'acquisition

Le système TEAM propose une phase d'acquisition sophistiquée, à travers une bonne interface graphique : la phase d'installation permet de compléter plusieurs composants du système en fonction des éléments de la base de données, décrits par l'administrateur de cette base. Précisément, cette phase affecte les composants suivants :

- **le lexique** : il contient les pronoms, conjonctions, déterminants, noms, adjectifs et verbes, etc. Dans la phase d'acquisition, ce lexique est enrichi des noms des champs et de leurs valeurs, ainsi que des mots communs désignant les fichiers. L'information associée à chaque mot est d'ordre syntaxique (catégorie, sous-catégorie, morphologie) et sémantique. L'information sémantique dépend de la catégorie syntaxique : ainsi les adjectifs et les verbes sont associés à un prédicat logique présent dans le modèle conceptuel, auquel ils font référence. Plus précisément, une correspondance est établie entre les constituants syntaxiques de la phrase et les arguments du prédicat.
- **Le schéma conceptuel** : il représente les concepts du domaine de la base. Il est organisé selon une « hiérarchie de types » très proche

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

d'une organisation orientée objets. Chaque champ de la base est associé aux objets de la hiérarchie du modèle conceptuel qui pourraient intervenir dans ce champ. Par ailleurs un ensemble de prédicats permet d'associer chaque objet du schéma conceptuel à un objet du schéma de la base de donnée. Dans la phase d'acquisition ce modèle est enrichi de noms, adjectifs, verbes, etc.

– **Le schéma de la base de données** : il est destiné à la traduction de la formule logique obtenue à partir de la question vers une requête à la base de données (en langage SODA). Il contient quatre types d'informations, la définition des objets du schéma conceptuel en termes de fichiers et de champs, la liste de synonymes pour chaque objet, une représentation logique de la structure de la base, et la liste des champs clefs des tables. C'est l'administrateur de la base de données qui va indiquer toutes ces informations.

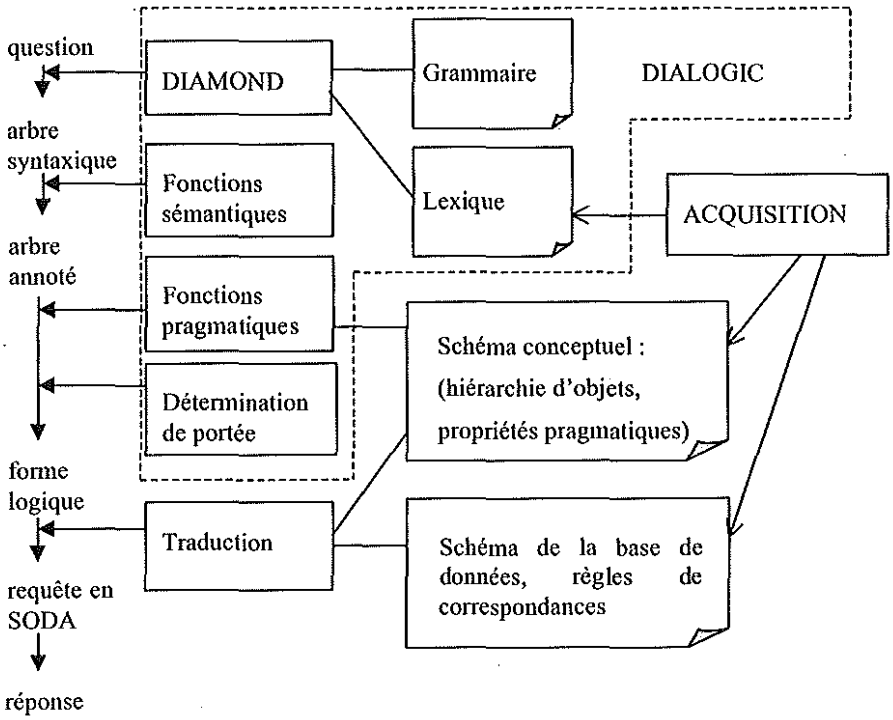

Figure 1 : Le fonctionnement général de TEAM

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

La *Figure 1* permet de situer le rôle de ces trois composants dans le système global. Tous sont conçus dans un formalisme du type logique du premier ordre, étendu par certains opérateurs intentionnels et d'ordre supérieur, et par des quantificateurs spéciaux.

2.3. Phase de fonctionnement

Le système qui traite une question en langue naturelle, appelé DIALOGIC, comprend les composants donnés *Figure 1*. La question est transformée par une analyse syntaxique (DIAMOND, système de règles qui exploite la grammaire et le lexique) en un arbre syntaxique. Cet arbre est ensuite annoté à l'aide des diverses fonctions d'interprétation sémantique, puis transformé par les fonctions pragmatiques, qui s'attachent en particulier au problème de la portée des quantificateurs. Ces fonctions exploitent le schéma conceptuel. La forme logique obtenue est optimisée, avant d'être traduite en une formule en SODA, langage de la base de données. Les règles de traduction s'appuient sur les définitions du schéma conceptuel et celles du schéma de la base de données.

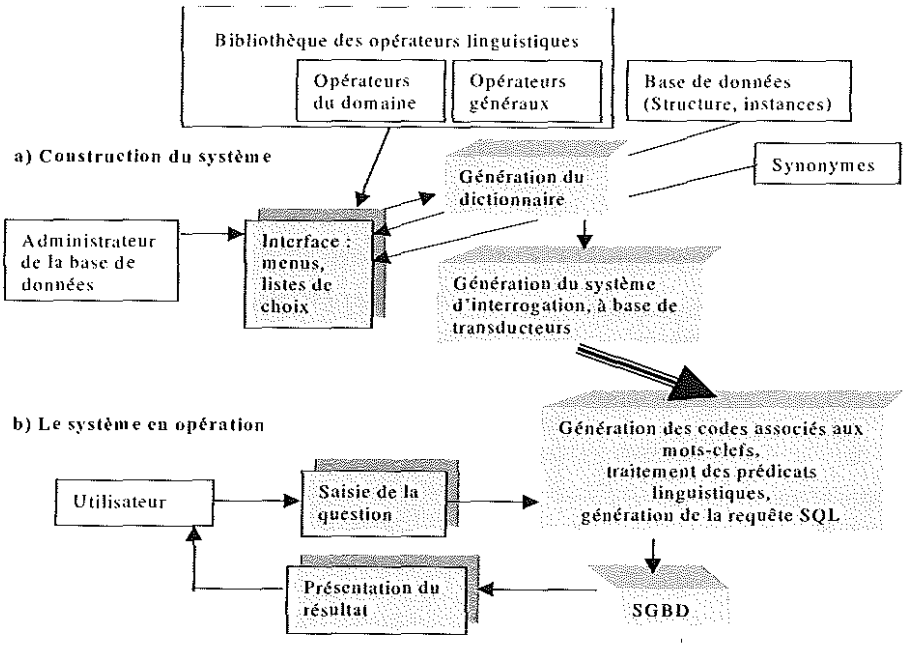

Figure 2 : Présentation générale de notre système

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

En résumé, TEAM est un système complet et sophistiqué, qui répond bien au but de transportabilité qui était celui de sa conception. La rapide présentation que nous venons d'en faire laisse assez entrevoir également sa complexité...

3. Approches par prédicats linguistiques

Nous donnons sur la *Figure 2* un aperçu de l'application que nous développons, laquelle se divise en deux classes d'outils : ceux qui « installent » le module d'interrogation sur la base de donnée (partie *a* de cette figure) et ceux qui réalisent l'interrogation (partie *b*).

3.1. L'installation du logiciel

L'utilitaire de construction du dictionnaire exploite :

- La structure de la base, présente dans un fichier créé par le système de gestion de bases de données (SGDB) utilisé. L'administrateur de cette base est sollicité pour associer, à chaque table et à chaque attribut, un mot français les représentant (et des synonymes, à partir d'un dictionnaire électronique de synonymes, inspiré de Gross 1998). Cette étape fait appel aux connaissances linguistiques normales d'un locuteur du français, et ne nécessite pas l'intervention d'un expert (ni de l'IA, ni de la linguistique).
- Le contenu de la base, c'est-à-dire toutes les instances qui y sont stockées, grâce aux techniques récentes permettant de construire directement un automate minimal (Mihov 1998), de le transformer en transducteur minimal (Mohri 1994) et de le compacter (Liang 1983). Les automates à nombre fini d'états, déjà promus par (Gross et Perrin 1989), sont d'ailleurs utilisés dans plusieurs domaines relevant du traitement du langage naturel (Roche et Schabes 1997).
- Des bibliothèques de prédicats linguistiques (les « opérateurs » de Z.S. Harris (Harris 1968, 1976). Nous travaillons à partir des travaux du LADL (en particulier Gross 1975 et Boons *et al.* 1976). Une de ces bibliothèques est générale, les autres correspondent au domaine de la base (économie, stock, tourisme...). La sémantique utilisée est celle des classes d'objets du LLI (Le Pesant et Mathieu-Colas 1998).

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

3.2. L'interrogation de la base

Une fois l'installation terminée, le système est prêt à fonctionner, selon le cycle :

1. Lecture de la question ;
2. Interprétation ;
3. Génération du code SQL [1] ;
4. Présentation du résultat.

Les étapes 1 à 3 sont exécutées par une cascade de transducteurs : le premier (étape 1) reconnaît les mots présents dans le dictionnaire et les remplace par les informations correspondantes. Le suivant (étape 2) travaille à partir des résultats du premier et met en œuvre les opérateurs d'interrogation de la base. Les derniers (étape 3) génèrent le code SQL d'interrogation de la base.

4. Les prédicats linguistiques

Afin d'unifier la présentation, tous les exemples ont trait à une base de données proposée dans l'article sur TEAM, base dont la structure est représentée *Figure 3*. Nous avons souligné sur cette figure les attributs qui représentent les tables elles-mêmes, c'est-à-dire, en général, le nom des lignes. Ces attributs particuliers seront indiqués au système par l'administrateur de cette base, lors de l'installation du logiciel.

Table-Pays : Nom-Pays : *chaîne*, Continent-Pays : *chaîne*, Population-Pays : *entier*, Capitale-Pays : *chaîne*, Superficie-Pays : *entier*

Table-Continent : Nom-Continent : *chaîne*, Hémisphère-Continent : *caractère*, Superficie-Continent : *entier*, Population-Continent : *entier*

Table-Capitale : Nom-Capitale : *chaîne*, Pays-Capitale : *chaîne*, Population-Capitale : *chaîne*

Table-Pic : Nom-Pic : *chaîne*, Pays-Pic :*chaîne*, Hauteur-Pic : *entier*, Volcan-Pic : *booléen*

Figure 3 : structure de la base utilisée en exemple

[1] SQL est le langage d'interrogation de bases de données le plus répandu aujourd'hui.

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

4.1. Les attributs

Lorsqu'on examine, en linguiste, les différents attributs d'une base de données, on s'aperçoit qu'ils correspondent à des verbes, noms ou adjectifs prédicatifs. D'après Harris, ceux-ci peuvent être mis en correspondance avec des phrases simples comportant, soit un verbe prédicatif, soit un verbe support approprié. Donnons quelques exemples :

1. Les quatre champs *Nom-Pays*, *Nom-Continent*, *Nom-Capitale* et *Nom-Pic* correspondent, entre autre, au verbe prédicatif *nommer* ou au nom prédicatif *nom*, par exemple dans les phrases simples :

   Ce(tte) (pays + continent + ville + pic) (a pour nom + est nommé(e) + est(le + la + l')) *Nom-(Pays + Continent + Capitale + Pic)*

2. De même, les champs *Population-Pays* et *Population-Capitale* sont associés au verbe prédicatif *habiter* ou au nom prédicatif *population* [1] :

   Ce(tte) (pays + ville) (a pour population + est habité par) *Population-(Pays + Capitale)* habitants

3. Quant au champ booléen *Volcan-Pic*, il est en correspondance avec les noms prédicatifs *volcan* et *cratère*, ainsi que l'adjectif prédicatif *volcanique* :

   Ce pic est (un volcan + volcanique + un cratère)

Ces prédicats linguistiques sont, toujours d'après Harris, des opérateurs sur les phrases simples. Ils ont pour arguments le sujet du verbe et ses compléments essentiels. L'exemple le plus simple d'interrogation consistera à extraire cette structure prédicative de la question :

4. Le Vulcano est-il un volcan?

Le premier transducteur remplace les mots-clefs [1] *Vulcano* et *vol-*

[1] En allemand, ces structures prédicatives peuvent s'exprimer, de manière identique, par des phrases simples, comme :

Dieses Land trägt den Namen ((heißt+ist) *Nom-Pays* genannt)+(ist Nom-Pays)
Dieses Land (hat *Population-Pays* Einwoher)+(istvon *Population-Pays* Menschen bewohnt)).

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

can par les informations stockées dans le dictionnaire électronique : *Vulcano* est une instance de la table *Table-Pic* et *volcan* un nom prédicatif associé à l'attribut *Volcan-Pic* de cette même table :

I.Table-Pic.Nom-Pic.Vulcano
A.Table-Pic.Volcan-Pic

Le deuxième transducteur organise l'interrogation : nous souhaitons connaître l'instance du champ *Volcan-Pic* lorsque le champ *Nom-Pic* est instancié par *Vulcano* :

A.Table-Pic.Volcan-Pic
/I.Table-Pic.Nom-Pic.Vulcano

Enfin, les transducteurs suivants réécrivent cet énoncé en une requête SQL, c'est-à-dire en trois champs, SELECT, FROM et WHERE :

SELECT [Table-Pic].[Nom-Pic],[Table-Pic].[Volcan-Pic]
FROM [Table-Pic]
WHERE ((([Table-Pic].[Nom-Pic])="Vulcano") AND(([Table-Pic].[Volcan-Pic])=True));

4.2. L'opérateur *DE*

Un opérateur très fréquent est celui qui résulte de la transformation de deux phrases simples en un complément prépositionnel en *de* :

1. Ce pays est nommé France # Ce pays a pour population soixante millions d'habitants

   La population de la France est de soixante millions d'habitants.

Ce complément prépositionnel correspondra, dans notre système, à un opérateur *DE* à deux arguments, qui sont les attributs (éventuellement instanciés) d'une même table. L'un de ces deux attributs doit correspondre à l'attribut représentant la table (l'attribut souligné sur la *Figure 3*). Un seul de ces attributs soulignés étant présent ici, le transducteur 2 choisit la table *Table-Pays* et non la table *Table-Capitale* (la table *Table-Pic* est éliminée car elle n'est présente que d'un seul côté

[1] Les autres sont ignorés. Les mots-clefs sont, soit des opérateurs, soit des attributs ou des instances. Ceux-ci sont traduits par une suite de codes (séparés par des points) : A (attribut) ou I (instance), le nom de la table, le nom de l'attribut et, éventuellement, l'instance.

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

de l'opérateur) :

2. Quelle est la population de la France[1] ?

(A.Table-Pays.Population-Pays+A.Table-Capitale.Population-Capitale)
DE
(I.Table-Pays.Nom-Pays.France+I.Table-Capitale.Pays-Capitale.France+I.Table-Pic.Pays-Pic.France)

DE(A.Table-Pays.Population-Pays, I.Table-Pays.Nom-Pays.France)

SELECT [Table-Pays].[Population-Pays]
FROM [Table-Pays]
WHERE ((([Table-Pays].[Nom-Pays])="France"));

Cet opérateur peut être évidemment répété, comme dans la question 0, où le mot *capitale* est reconnu, soit comme le nom de la table *Table-Capitale* (et remplacé par l'attribut *Nom-Capitale*), soit comme l'attribut *Capitale-Pays* de la table *Table-Pays*. Lors de la recherche des arguments du premier opérateur, il sélectionne *A.Table-Capitale.Nom-Capitale*, ce qui contraint la deuxième recherche à la table *Table-Capitale*[2].

3. Quelle est la population de la capitale de la France?

(A.Table-Pays.Population-Pays+A.Table-Capitale.Population-Capitale)
DE
(A.Table-Capitale.Nom-Capitale+A.Table-Pays.Capitale-Pays)
DE
(I.Table-Pays.Nom-Pays.France+I.Table-Capitale.Pays-Capitale.France+I.Table-Pic.Pays-Pic.France)

DE(A.Table-Capitale.Population-Capitale, A.Table-Capitale.Nom-Capitale)
/DE(A.Table-Capitale.Nom-Capitale,I.Table-Capitale.Pays-Capitale.France)

---

[1] En allemand, un génitif peut remplacer la préposition. Mais, quoiqu'il en soit, le transducteur produira la même sortie, l'opérateur *DE* : Was ist die Bevölkerung (von Frankreich+Frankreichs).

[2] Sinon, le deuxième groupe nominal aurait pu donner : DE(A.Table-Pays.Capitale-Pays, I.Table-Pays.Nom-Pays.France).

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

SELECT[Table-Capitale].[Population-Capitale]
FROM [Table-Capitale] INNER JOIN [Table-Pays] ON[Table-Capitale].[Pays-Capitale] = [Table-Pays].[Nom-Pays]
WHERE ((([Table-Pays].[Nom-Pays])="France"));

4.3. L'opérateur SUP

D'autres questions nécessiteront l'introduction d'un opérateur particulier, dont nous stockerons la traduction SQL, donnée par l'administrateur de la base. Prenons pour exemple l'opérateur *SUP*. Celui-ci appartient à la bibliothèque commune à tous les domaines où il est noté comme un opérateur à deux arguments de type numérique. Notre programme d'installation, rappelons-le, consulte le schéma de la base et va donc proposer à l'administrateur quatre versions de cet opérateur :

1. SUP(A.Table-Pays.Population-Pays, A.Table-Pays.Population-Pays)
   SUP(A.Table-Pays.Superficie-Pays, A.Table-Pays.Superficie-Pays)
   SUP(A.Table-Capitale.Population-Capitale,A.Table-Capitale.Population-Capitale)
   SUP(A.Table-Pic.Hauteur-Pic, A.Table-Pic.Hauteur-Pic)[1]

La dernière ligne correspond, par exemple, à la forme comparative de l'adjectif prédicatif *haut* :

Un pic est plus haut qu'un pic

Elle sera traduite en SQL par l'administrateur, ce qui va permettre une question comme :

---

[1] En fait, l'installation propose, dans le même temps, quatre autres versions de cet opérateur *SUP* :

SUP(A.Table-Pays.Population-Pays, Dnum)
SUP(A.Table-Pays.Superficie-Pays, Dnum)
SUP(A.Table-Capitale.Population-Capitale, Dnum)
SUP(A.Table-Pic.Hauteur-Pic, Dnum)

destinées à répondre à des questions comme :

Quels sont les sommets de plus de 1000 mètres ?

Pour alléger l'exposé, nous n'en ferons pas mention par la suite, puisqu'ils ne correspondent pas à nos exemples.

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

2. Le Mont-Dore est-il plus haut que le Vulcano[1] ?

I.Table-Pic.Nom-Pic.Mont-Dore
SUP(A.Table-Pays.Population-Pays, A.Table-Pays.Population-Pays)
+SUP(A.Table-Pays.Superficie-Pays, A.Table-Pays.Superficie-Pays)
+SUP(A.Table-Capitale.Population-Capitale,A.Table-Capitale.Population-Capitale)
+SUP(A.Table-Pic.Hauteur-Pic,A.Table-Pic.Hauteur-Pic)
A.Table-Pic.Hauteur-Pic
I.Table-Pic.Nom-Pic.Vulcano

SUP(A.Table-Pic.Hauteur-Pic, A.Table-Pic.Hauteur-Pic)
/I.Table-Pic.Nom-Pic.Mont-Dore
/I.Table-Pic.Nom-Pic.Vulcano

Dans la formule SQL, trois étapes sont nécessaires, calculer la hauteur des deux pics, chercher le plus grand des deux et son nom :

```
CREATE VIEW [Deux Pics] AS
SELECT [Table-Pic].[Nom-Pic], [Table-Pic].[Hauteur-Pic]
FROM [Table-Pic]
WHERE ((([Table-Pic].[Nom-Pic])="Vulcano" Or([Table-
      Pic].[Nom-Pic])="Mont-Dore"));
CREATE VIEW [Max 2 Pics] AS
SELECT Max([Deux Pics].[Hauteur-Pic]) AS [MaxDeHauteur-Pic]
FROM [Deux Pics];
SELECT [Deux Pics].[Nom-Pic]
FROM [Deux Pics] INNER JOIN [Max 2 Pics] ON [Deux
      Pics].[Hauteur-Pic] =[Max 2 Pics].[MaxDeHauteur-Pic];
```

4.4. L'opérateur MAX

Étudions maintenant la structure prédicative associée à la forme superlative de l'adjectif prédicatif *haut* :

[1] Le comparatif allemand transforme l'adjectif *hoch* en *höher*. Les deux formes seront enregistrées lors de la constitution du dictionnaire, pendant l'installation du logiciel. Dans la question :

Ist der Mont-Dore höher als der Vulcano ?

En lisant le mot *höher* le transducteur 1 générera directement l'opérateur *SUP* et ses arguments :*SUP(A.Table-Pic.Hauteur-Pic, A.Table-Pic.Hauteur-Pic).*

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

1. Un sommet est le plus haut de tous les sommets #. Ces sommets sont en France.

Un sommet est le plus haut de tous les sommets de France
Un sommet est le plus haut de France

L'opérateur considéré ici fait référence à un calcul sur l'attribut *Hauteur-Pic* de la table *Table-Pic* (représentée par son nom, l'attribut souligné *Nom-Pic*) lorsqu'un autre attribut de cette même table, l'attribut *Pays-Pic*, est instancié à *France* :

2. Quel est le plus haut sommet de France[1] ?

MAX(A.Table-Pays.Population-Pays)
+MAX(A.Table-Pays.Superficie-Pays)
+MAX(A.Table-Capitale.Population-Capitale)
+MAX(A.Table-Pic.Hauteur-Pic)
A.Table-Pic.Hauteur-Pic
A.Table-Pic.Nom-Pic
DE
(I.Table-Pays.Nom-Pays.France+I.Table-Capitale.Pays-Capitale.France+I.Table-Pic.Pays-Pic.France)

A.Table-Pic.Nom-Pic
/MAX(A.Table-Pic.Hauteur-Pic)
/I.Table-Pic.Pays-Pic.France

L'interrogation en SQL se déroule en deux phases, la hauteur maximum est tout d'abord stockée dans une table (nommée *Maxi*) :

```
CREATE VIEW [Maxi] AS
SELECT [Table-Pays].[Nom-Pays], Max([Table-Pic].[Hauteur-Pic])
AS[MaxDeHauteur-Pic]
      FROM [Table-Pic] INNER JOIN [Table-Pays] ON [Table-
            Pic].[Pays-Pic] =[Table-Pays].[Nom-Pays]
GROUP BY [Table-Pays].[Nom-Pays]
HAVING ((([Table-Pays].[Nom-Pays])="France"));
```

Puis on cherche le nom du sommet (ou des sommets) correspondant à cette hauteur :

[1] Comme pour le comparatif, le choix de l'opérateur *MAX* et de son argument *A.Table-Pic.Hauteur-Pic* proviendra d'un seul mot, le mot *höchste* :
Was ist der höchste Gipfel (von Frankreich+Frankreichs).

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

```
SELECT [Maxi].[MaxDeHauteur-Pic],[Table-Pic].[Nom-Pic]
FROM [Table-Pic] INNER JOIN [Maxi] ON [Table-
    Pic].[Hauteur-Pic] =[Maxi].[MaxDeHauteur-Pic];
```

5. Trois exemples plus élaborés

Considérons maintenant trois questions présentées et discutées dans TEAM :

1. Donne le plus haut pic de chaque continent
2. Donne le plus haut pic de chaque pays d'Asie
3. Donne le plus haut pic des pays d'Asie

Les tableaux de la *Figure 4* et de la *Figure 5* présentent, pour chacune des questions 1 et 2 ci-dessus, le résultat obtenu après le passage du premier transducteur.

5.1. Donne le plus haut pic de chaque continent

Donne	
le plus	MAX(A.Table-Pays.Population-Pays) MAX(A.Table-Pays.Superficie-Pays) MAX(A.Table-Capitale.Population-Capitale) MAX(A.Table-Pic.Hauteur-Pic)
haut	A.Table-Pic.Hauteur-Pic
pic	A.Table-Pic.Nom-Pic
de	DE DANS(A.Table-Capitale.Nom-Capitale, A.Table-Continent.Nom-Continent) DANS(A.Table-Pic.Nom-Pic,A.Table-Continent.Nom-Continent)
chaque	
continent	A.Table-Continent.Nom-Continent

Figure 4 :
La première question après le passage du premier transducteur

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

Le deuxième transducteur lit, de gauche à droite, les quatre versions de l'opérateur *MAX* et sélectionne la dernière, à cause de la présence de l'adjectif prédicatif *haut*, traduit par l'attribut *A.Table-Pic.Hauteur-Pic*. On recherche un pic, c'est-à-dire l'attribut *A.Table-Pic.Nom-Pic*. Nous obtenons donc :

A.Table-Pic.Nom-Pic
/MAX(A.Table-Pic.Hauteur-Pic)

Puis vient l'interprétation de la fin de la question, *pic de chaque continent* : comme deux tables différentes, les tables *Table-Pic* et *Table-Continent*, sont mises en relation, le *de* ne représente pas l'opérateur *DE*, interne à une table, mais un autre opérateur, l'opérateur *DANS* qui est défini par des jointures entre tables ; ces jointures sont, rappelons-le, déclarées par le gestionnaire de la base lors de l'installation. Deux versions existent dans notre exemple de base :

DANS(A.Table-Capitale.Nom-Capitale,A.Table-Continent.Nom-Continent)
DANS(A.Table-Pic.Nom-Pic, A.Table-Continent.Nom-Continent)

C'est la deuxième qui correspond à notre contexte(*pic* et *continent*, ou, plutôt, *A.Table-Pic.Nom-Pic* et *A.Table-Continent.Nom-Continent*. Comme aucun attribut n'est instancié, nous obtenons, *a priori*, une liste :

DANS(A.Table-Pic.Nom-Pic,A.Table-Continent.Nom-Continent)

Les deux conditions se coordonnent finalement pour donner :

DANS(A.Table-Pic.Nom-Pic,A.Table-Continent.Nom-Continent)
/MAX(A.Table-Pic.Hauteur-Pic)

Il y a ensuite « traduction » de ces opérateurs en une requête SQL, présentée en 8.1., Annexe 1.

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

5.2. Donne le plus haut pic de chaque pays d'Asie

Donne	
le plus	MAX(A.Table-Pays.Population-Pays) MAX(A.Table-Pays.Superficie-Pays) MAX(A.Table-Capitale.Population-Capitale) MAX(A.Table-Pic.Hauteur-Pic)
haut	A.Table-Pic.Hauteur-Pic
pic	A.Table-Pic.Nom-Pic
de	DE DANS(A.Table-Capitale.Nom-Capitale, A.Table-Continent.Nom-Continent) DANS(A.Table-Pic.Nom-Pic,A.Table-Continent.Nom-Continent)
chaque	
pays	A.Table-Pays.Nom-Pays A.Table-Capitale.Pays-Capitale A.Table-Pic.Pays-Pic
d'	DE DANS(A.Table-Capitale.Nom-Capitale, A.Table-Continent.Nom-Continent) DANS(A.Table-Pic.Nom-Pic, A.Table-Continent.Nom-Continent)
Asie	I.Table-Pays.Continent-Pays.Asie I.Table-Continent.Nom-Continent.Asie

Figure 5 :
La deuxième question après le passage du premier transducteur

La première partie de cette question est identique à la précédente :

A.Table-Pic.Nom-Pic
/MAX(A.Table-Pic.Hauteur-Pic)

Puis viennent, cette fois-ci, deux opérateurs *DE* : le *de* met en relation deux attributs de la table *Table-Pic*, A.Table-Pic.Nom-Pic et A.Table-Pic.Pays-Pic, et le *d'*, deux attributs de la table *Table-Pays*,

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

A.Table-Pays.Nom-Pays etI.Table-Pays.Continent-Pays.Asie. Comme le premier opérateur *DE* n'est pas instancié, le résultat est une liste :

DE(A.Table-Pic.Nom-Pic,A.Table-Pic.Pays-Pic)
/DE(A.Table-Pays.Nom-Pays,I.Table-Pays.Continent-Pays.Asie)

Ce qui nous donne finalement :

DE(A.Table-Pic.Nom-Pic,A.Table-Pic.Pays-Pic)
/DE(A.Table-Pays.Nom-Pays,I.Table-Pays.Continent-Pays.Asie)
/MAX(A.Table-Pic.Hauteur-Pic)

La formule SQL correspondante est donnée en 8.2., Annexe2.

5.3. Donne le plus haut pic des pays d'Asie

A priori, cette question est presque la même que la précédente, mais la réponse ne doit plus être une table des sommets les plus haut de chaque pays d'Asie, mais le sommet le plus haut de cette table. En fait, en y regardant de plus près, on s'aperçoit que les transformations n'opèrent pas sur les mêmes phrases simples. Pour la question 3, pays est un singulier :

1. Un sommet est le plus haut de tous les sommets # Ces sommets sont dans un pays # Ce pays est en Asie

   Un sommet est le plus haut de tous les sommets de chaque pays d'Asie.

Alors que, pour la question 0, pays est un pluriel :

2. Un sommet est le plus haut de tous les sommets # Ces sommets sont dans des pays # Ces pays sont en Asie

   Un sommet est le plus haut de tous les sommets des pays d'Asie.

Comment distinguer entre les deux? Il nous semble, tout simplement, que nous devons avoir deux opérateurs, *MAX* et *MAX!*, repérables par leur trace respective *le plus_de* et *le plus_des*. Nous aurons donc ici :

DE(A.Table-Pic.Nom-Pic,A.Table-Pic.Pays-Pic)
/DE(A.Table-Pays.Nom-Pays,I.Table-Pays.Continent-Pays.Asie)
/MAX!(A.Table-Pic.Hauteur-Pic)

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

qui se traduira non plus par une liste, mais par un seul élément. Nous présentons en 8.3., Annexe 3 son codage SQL.

5.4. La réponse de TEAM à la question 1

Pour terminer, nous allons donner ci-dessous la réponse de TEAM à la question 1. Cet exemple y est analysé de la façon suivante :

- La phase d'analyse syntaxique génère quatre arbres syntaxiques, correspondant à quatre interprétations distinctes. L'un de ces arbres est ensuite sélectionné grâce au contexte, à l'aide des fonctions sémantiques.
- Les relations du schéma conceptuel (comme par exemple (*PK-CONTINENT-OF* PEAK5 CONTINENT2) et (*PEAK* PEAK1)) sont associées aux nœuds de l'arbre.
- Les « fonctions pragmatiques » vont travailler sur ces informations, en changeant par exemple le génitif « continent's » en « PK-CONTINENT-OF ». Cela va générer une première forme logique de la question.
- Puis un autre système de règles de réécriture va intervenir pour aboutir à une forme logique simplifiée, plus adaptée au langage d'interrogation de base de données utilisé pour cible (*SODA*).
- C'est cette forme logique simplifiée qui va être traduite en une formule SODA par le *schema translator* qui va :
  - traduire les divers opérateurs et quantificateurs logiques en opérateurs d'interrogation ;
  - traduire les prédicats et leurs arguments en éléments précis de la base de données : table, champ ou valeur ;
  - déterminer quelles informations inclure dans la réponse ;
  - éliminer les contraintes redondantes dans la forme logique (en deux phases : une expansion et une simplification).

La formule complète obtenue en SODA est donnée en 8.4., Annexe 4.

Nous pensons avoir conçu un système plus simple, permettant une réponse rapide et pertinente aux mêmes questions.

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

6. Perspectives de recherche

Une première version de notre système a été implémentée dans le cadre plus restreint d'une base de données géographique (Thomas *et al.* 1997). Il nous a permis de nous rendre compte d'un certain nombre d'insuffisances que nous avons cherché à résoudre par la réflexion théorique que nous venons de présenter. Un prototype plus général est en cours d'élaboration. Nous espérons pouvoir bientôt vérifier par l'expérience les exemples que nous venons de discuter.

7. Références

BOONS J.-P., GUILLET A., LECLERE C. : 1976a, *La structure des phrases simples en français.* I. *Constructions intransitives* (Genève : Droz).

BOONS J.-P., GUILLET A., LECLERE C. : 1976b, *La structure des phrases simples en français.* II. *Constructions transitives* (Paris : Rapport de recherche du LADL, n°6).

CLIFFORD J. : 1988, « Natural Language Querying of Historical Databases », *Computational Linguistics*, 14-4, p. 10–34.

GREEN B.F., WOLF A.K., CHOMSKY C., LAUGHERY K. : 1960, « Baseball : an automatic question-answerer », *Computers and thought* (New-York : Mc Graw Hill), p. 207–216.

GROSS M., PERRIN D. : 1989, « Electronic Dictionaries and Automata in Computational Linguistics », *Lecture Notes in Computer Science*, 377, Springer.

GROSS M. : 1975, *Méthodes en syntaxe* (Paris : Hermann).

GROSS G. : 1998, « Pour une véritable fonction synonymie dans un traitement de texte », *Langages*, 131, p. 103–114.

GROSZ B., APPELT D., MARTIN P., PEIRERA F. : 1987, « TEAM : an experiment in the design of transportable natural language interfaces », *Artificial Intelligence*, 32, p. 173–243.

HARRIS Z.S. : 1968, *Mathematical Structures of Language* (Interscience Publishers).

HARRIS Z.S. : 1976, *Notes du cours de syntaxe* (Paris : Le Seuil).

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

KAPLAN S.J. : 1984, « Designing a Portable Natural Language Database Query System », *ACM Transactions on Database Systems*, 9,1, march 1984, p. 1–19.

LE PESANT D., MATHIEU-COLAS M. : 1998, « Introduction aux classes d'objets », *Langages*, 131, p. 6–33.

LIANG F.M. : 1983, *Word Hyphenation by Computer*, PhD Thesis (Computer Science Departement, Standford University, Research Report STAN-CS-83-977).

MIHOV S. : 1998, *On-line Algorithm for Building Minimal Automaton Presenting Finite Language* (Annuaire de l'Université de Sofia St. Kl. Ohridski, Faculté de Mathématiques et Informatique, 91, 1).

MOHRI M. : 1994, « Minimization of Sequential Transducers », *LNCS:807*.

PEQUEGNAT C., AGUIRRE J.L. : 1988, « Interface en langue naturelle pour l'interrogation d'une base de données relationnelle », *Systèmes experts et applications*, pp. 441–460.

ROCHE E., SCHABES Y. : 1997, eds. *Finite state language Processing* (Cambridge, Mass./London, England : MIT Press).

SABAH G. : 1997, « Le sens dans le traitement automatique des langues », *T.A.L.*, 38,2, p. 91–133.

THOMAS N., MAUREL D., TREPIED C. : 1997, *Interrogation en langage naturel d'une base de données spatiales*, Rapport interne du LI, n° 190, E3i-Université de Tours.

8. Annexes

8.1. Annexe 1

Requêtes SQL pour « **Donne le plus haut pic de chaque continent.** »

1. Recherche de la hauteur maximale pour chaque continent, le résultat est une table nommée « *Maxi* » :

```
CREATE VIEW [Maxi] AS
SELECT [Table-Continent].[Nom-Continent],Max([Table-Pic].[Hauteur-
    Pic]) AS [MaxDeHauteur-Pic]
```

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

```
FROM ([Table-Pic] INNER JOIN [Table-Pays] ON[Table-Pic].[Pays-Pic] =
    [Table-Pays].[Nom-Pays]) INNER JOIN [Table-Continent] ON
    [Table-Pays].[Continent-Pays]= [Table-Continent].[Nom-Continent]
GROUP BY [Table-Continent].[Nom-Continent];
```

2. recherche des noms des pics de hauteur maximale pour chaque continent, qui utilise la table nommée « *Maxi* » :

```
SELECT [Maxi].[Nom-Continent], [Table-Pic].[Nom-Pic],
    [Maxi].[MaxDeHauteur-Pic]
FROM [Maxi] INNER JOIN [Table-Pic] ON[Maxi].[MaxDeHauteur-Pic] =
    [Table-Pic].[Hauteur-Pic]
GROUP BY [Maxi].[Nom-Continent], [Table-Pic].[Nom-
    Pic],[Maxi].[MaxDeHauteur-Pic];
```

8.2. Annexe 2

Requêtes SQL pour « **Donne le plus haut pic de chaque pays d'Asie.** »

```
CREATE VIEW [Maxi] AS
SELECT [Table-Pays].[Continent-Pays],[Table-Pays].[Nom-Pays],
    Max([Table-Pic].[Hauteur-Pic]) AS[MaxDeHauteur-Pic]
FROM [Table-Pic] INNER JOIN [Table-Pays] ON[Table-Pic].[Pays-Pic] =
    [Table-Pays].[Nom-Pays]
GROUP BY[Table-Pays].[Continent-Pays], [Table-Pays].[Nom-Pays]
HAVING((([Table-Pays].[Continent-Pays])="Asie"));
SELECT [Maxi].[Nom-Pays], [Table-Pic].[Nom-
    Pic],[Maxi].[MaxDeHauteur-Pic]
FROM [Maxi] INNER JOIN [Table-Pic] ON[Maxi].[MaxDeHauteur-Pic] =
    [Table-Pic].[Hauteur-Pic];
```

8.3. Annexe 3

Requêtes SQL pour « **Donne le plus haut pic des pays d'Asie.** »

```
CREATE VIEW [Maxi] AS
SELECT [Table-Pays].[Continent-Pays], Max([Table-Pic].[Hauteur-Pic])
    AS[MaxDeHauteur-Pic]
FROM [Table-Pic] INNER JOIN [Table-Pays] ON[Table-Pic].[Pays-Pic] =
    [Table-Pays].[Nom-Pays]
GROUP BY [Table-Pays].[Continent-Pays]
```

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.

```
HAVING((([Table-Pays].[Continent-Pays])="Asie"));
SELECT [Table-Pic].[Nom-Pic], [Maxi].[MaxDeHauteur-Pic]
FROM [Maxi] INNER JOIN [Table-Pic] ON[Maxi].[MaxDeHauteur-Pic] =
      [Table-Pic].[Hauteur-Pic];
```

8.4. Annexe 4

Formule SODA pour « **Donne le plus haut pic de chaque continent.** »

```
((IN #:$1TABLE-CONTINENT) (MAX (#:$3 HAUTEUR-PIC)
                                (IN#:$2 TABLE-PAYS)
                                (IN #:$3 TABLE-PIC)
                                ((#:$3 PAYS-PIC) EQ (#:$2 NOM-PAYS))
                                ((#:$2 CONTINENT-PAYS) EQ (#:$1
                                      NOM-CONTINENT))
                           (?(#:$1 NOM-CONTINENT))
                           (? (#:$3 HAUTEUR-PIC))
                           (? (#:$3 NOM-PIC))
```

Extrait de la Revue Informatique et Statistique dans les Sciences humaines
XXXV, 1 à 4, 1999. C.I.P.L. - Université de Liège - Tous droits réservés.