Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

Centre Génie Industriel et Informatique (G2I)

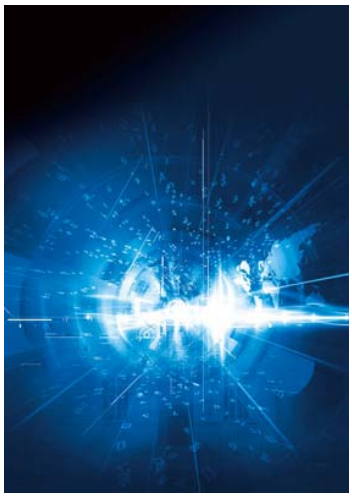
TRAITEMENT AUTOMATIQUE DU LANGAGE NATUREL POUR L'EXTRACTION ET LA RECHERCHE D'INFORMATIONS

X. TANNIER

Mars 2006

RAPPORT DE RECHERCHE

2006-400-006



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

Table des matières

Index

Introduction

Ce document est la version longue d'un chapitre d'état de l'art d'une thèse concernant l'extraction et la recherche d'information en langage naturel dans les documents structurés. Nous avons pensé qu'il pouvait être utile en lui-même aux personnes désireuses de se familiariser avec les problématiques de l'analyse des documents en langage naturel.

Cet état de l'art n'est donc en aucun cas une description exhaustive des applications des techniques de traitement automatique des langues, mais une introduction à certaines problématiques, choisies au départ pour correspondre avec le sujet général de la thèse présentée.

Le domaine du langage naturel et de son traitement automatique se trouve au cœur de la problématique de l'extraction et de la recherche d'information. Il semble évident que les progrès futurs passeront par une meilleure "compréhension" de la langue. L'état actuel de la recherche est loin de cette compréhension, et de nombreuses difficultés se présentent à tous les niveaux de l'analyse de l'écrit. Les problèmes peuvent ainsi être d'ordres morphologique*, syntaxique*, sémantique* ou pragmatique*.

Dans un premier temps, de la section 1 à la section 4, ce document décrira ces différents niveaux d'étude de la langue, et abordera quelques-uns des écueils auxquels se heurte particulièrement toute tentative d'analyse automatique de textes.

Nous développerons par la suite les aspects du traitement automatique de la langue qui ont été explorés dans les systèmes de recherche d'information (section 5), puis nous adorderons le domaine des interfaces en langage naturel pour les bases de données (section 6). Enfin, nous donnerons un aperçu des enjeux et des premières approches concernant les interfaces en langage naturel pour la recherche d'information semi-structurée (section 7).

1 Les différents niveaux du langage

L'analyse du langage nécessite une connaissance de sa structure sur de nombreux niveaux : que sont les mots ? Que signifient-ils ? Comment se combinent-ils pour former la phrase ? Comment contribuent-ils au sens de la phrase ? Et, par ailleurs, comment fonctionnent le monde et le raisonnement de l'humain dans le monde ? Par exemple, pour participer à une conversation, un être humain doit connaître non seulement le langage utilisé, mais également les règles du monde autour duquel lui et son interlocuteur vivent, ainsi que les règles élémentaires de la conversation[1].

De la réception des sons (ou leur prononciation) jusqu'à la compréhension approfondie des mots prononcés dans l'environnement où ils sont prononcés, les linguistes distinguent plusieurs palliers permettant l'analyse ou la génération d'un énoncé en langage naturel. Ces niveaux de connaissance restent bien entendu toujours valables lorsque l'on aborde l'analyse *automatique* de la langue. Les grandes spécialités sont :

- la **phonétique** et la **phonologie**, ou comment les mots et les phrases sont liés aux sons qui les réalisent à l'oral [21, 59]. Ne traitant que l'écrit, nous ne reviendrons pas sur ce domaine.

morphologie

- la **morphologie**, ou comment les mots sont construits et quels sont leurs rôles dans la phrase [69].

[1] Il doit par exemple, selon *Grice* [47], respecter les quatre maximes de quantité (donner la quantité d'information nécessaire), de qualité (dire ce qu'on pense vrai), de pertinence (rester en relation avec le sujet de l'échange) et de manière (être clair et éviter l'ambiguïté).

– la **syntaxe**[*], ou comment les mots se combinent pour former des *syntagmes*[*], puis des *propositions* et enfin des *phrases* correctes. *syntaxe*
– la **sémantique**[*], ou comment les mots font du sens lorsqu'ils sont insérés dans une phrase (indépendamment du contexte) [52]. *sémantique*
– la **pragmatique**[*], ou comment les phrases peuvent être interprétées selon leur contexte d'énonciation (interlocuteurs, phrases précédentes, connaissance commune du monde, ...) [63, 27]. *pragmatique*

Allen [5] donne les exemples suivants pour faire la distinction entre syntaxe, sémantique et pragmatique. Considérons que les phrases suivantes sont candidates pour figurer en tête du présent mémoire, c'est-à-dire qu'elles sont énoncées en l'absence totale de contexte :

(1) Le domaine du langage naturel et de son traitement automatique se trouve au cœur de la problématique de l'extraction et de la recherche d'information.

(2) Les grenouilles vertes ont des gros nez.

(3) Les idées vertes ont des gros nez.

(4) Vertes des ont les idées nez gros.

La première phrase semble être un début raisonnable pour un tel rapport. Elle correspond à tout ce qui est connu en matière de syntaxe, de sémantique et de pragmatique. La phrase 2 est bien formée sur les plans syntaxique et sémantique, mais pas pragmatique. En effet, elle conviendrait mal comme première phrase d'un mémoire de thèse, et le lecteur ne verrait aucune raison valable de la voir utilisée.

Mais l'exemple 3 serait pire encore : il est à la fois pragmatiquement et sémantiquement mal formé. On remarque en effet qu'il est possible d'affirmer que la phrase 2 est vraie ou fausse, tandis que c'est impossible pour 3 dans une conversation cohérente. La structure en est pourtant correcte, mais des idées ne peuvent pas être vertes et, même si elles le peuvent dans certains contextes, elles n'ont certainement pas de nez, ni gros ni maigre[1].

Enfin, la phrase 4 est tout simplement inintelligible : elle contient pourtant les mêmes mots que la précédente, mais ne respecte aucune des structures grammaticales (syntaxiques) admises en Français.

Dans le cadre de notre travail, nous nous basons uniquement sur la langue écrite[2], ce qui implique que les entités les plus petites que nous allons étudier (les *unités grammaticales*) sont les mots. Cela ne signifie pourtant pas que la structure interne du mot, en particulier les différentes flexions[*], n'est pas prise en compte. Mais le mot comporte l'avantage d'être une unité relativement facile à distinguer dans un texte (en tout cas dans les langues française et anglaise, qui vont nous intéresser plus particulièrement), ce qui n'est pas le cas pour la langue orale.

[1] Cet exemple est une référence à la célèbre phrase *"Colorless green ideas sleep furiously"* de Noam Chomsky, et donne une bonne idée de la distinction entre syntaxe et sémantique. Pourtant de nombreux linguistes, notamment fonctionnalistes et cognitivistes [40], refutent la possibilité de trouver des phrases *asémantiques*. Selon eux, cette phrase, comme toutes les phrases syntaxiquement correctes, peut avoir un sens dans un contexte particulier. Ainsi Yuen Ren Chao écrivit en 1971 *"The Story of my Friend, Whose Colourless Green Ideas Sleep Furiously"*.

[2] L'analyse d'une question formulée à l'oral est pourtant aussi un aspect intéressant, notamment pour des raisons d'accessibilité.

2 Morphologie

La morphologie [68] est l'étude de la forme des mots (de leur flexion – indications de cas, genre, nombre, mode, temps, etc. – de leur dérivation – préfixes, suffixes, infixes – et de leur composition – mots composés). Sous l'appellation de morphosyntaxe, elle représente également l'étude des règles de combinaison des morphèmes (unités minimales de sens) selon la configuration syntaxique de l'énoncé [75].

En pratique, dans le cadre du traitement automatique de la langue, l'analyse morphologique consiste à segmenter le texte en unités élémentaires (*tokenisation*) et à déterminer les différentes caractéristiques de ces unités.

2.1 Segmentation du texte

> Mot *: Son ou groupe de sons articulés ou figurés graphiquement, constituant une unité porteuse de signification à laquelle est liée, dans une langue donnée, une représentation d'un être, d'un objet, d'un concept, etc.*
> Trésor de la Langue Française.

Pour passer d'un simple alignement de caractères à une quelconque représentation, un texte doit d'abord être segmenté en unités élémentaires et en phrases.

mot

Le *mot*, l'unité élémentaire qui semble évidente pour la construction d'un énoncé, n'a pas de définition simple en ce concerne l'analyse automatique. Définir une liste de caractères délimiteurs (l'espace, la ponctuation, l'apostrophe)[1] ne suffit pas à séparer les mots. Par exemple, l'apostrophe distingue souvent deux mots (en marquant une élision, comme dans *"le chat d'Alphonse"*) mais peut poser problème (en Français dans *"aujourd'hui"* par exemple, mais surtout en Anglais avec l'expression de la possession – *"Alphonse's cat"* ou *"the boys' toys"*). Le trait d'union est également ambigu sur ce point : il peut être interne à une unité ou avoir une fonction syntaxique spécifique dans l'ensemble[2]. Par ailleurs, on peut considérer que les unités élémentaires incluent potentiellement des espaces, comme dans *"pomme de terre"*.

Enfin la segmentation en phrases n'est pas triviale non plus. La présence d'une ponctuation forte, en particulier le point, n'est pas une garantie. En effet celui-ci a également la fonction d'indiquer les abréviations (*"M. le ministre"*, *"etc."*) ou le séparateur numérique en Anglais.

En pratique les systèmes qui mettent en place une segmentation du texte utilisent une liste de séparateurs par défaut, à laquelle ils ajoutent des connaissances lexicales et morphosyntaxiques pour traiter les cas ambigus[3,4].

2.2 Catégories grammaticales

catégorie grammaticale

L'information morphosyntaxique la plus importante est la catégorie grammaticale,

[1] On parle alors de *formes graphiques*, ou *tokens*.

[2] En Français, des exemples de séquences sont *"garde-fou"*, *"c'est-à-dire"*, *"qu'en-dira-t-on"* (avec une apostrophe en prime !), *"lui-même"*, *"arrive-t-il"*, *"puissé-je"*... En Anglais, le problème se pose de façon plus pressante encore, avec des constructions comme *"a text-based medium"*, *"the 90-cent-an-hour raise"*, *"26-year-old"* [65, chap 4].

[3] Ainsi, en Français, il est facile de lister les mots contenant une apostrophe, relativement rares (*"presqu'île"*, *"aujourd'hui"*, etc.). Quant au trait d'union, il n'est externe au mot que lorsqu'il est suivi d'un pronom préverbal.

[4] En ce qui concerne la segmentation en phrases, *Pappa et al.* ont récemment proposé un système de détection indépendant de la langue [77].

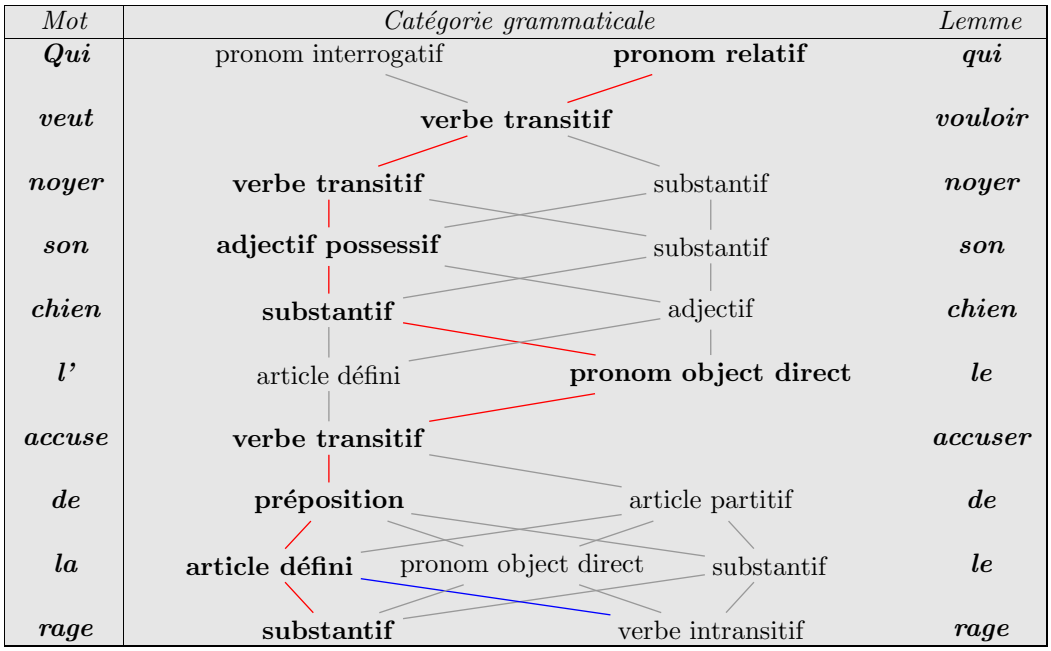


FIG. 1 – Catégories grammaticales possibles pour chacun des mots de la phrase *"Qui veut noyer son chien l'accuse de la rage"*. Les catégories correctes sont indiquées en gras. Les flexions* (voir plus bas) ne sont pas prises en compte (par exemple, *"accuse"* peut être au subjonctif ou à l'indicatif). Les diverses possibilités et le nom des catégories sont issus du *Trésor de la Langue Française*. A droite se trouvent les lemmes* dont dérivent les formes de la phrase.

qui regroupe sous une étiquette commune les mots partageant le même comportement syntaxique dans un énoncé. Le *nom*, le *verbe*, l'*adjectif* sont des catégories grammaticales. Le nombre de catégories n'est pas fixé, il est défini dans chaque application par un compromis entre complexité et spécificité[1]. La liste de catégories la plus utilisée (en Anglais) est la *Penn Treebank* [66] et compte 45 classes[2]. Celle du *Brown Corpus* [38] en répertorie 87, d'autres dépassent la centaine. Le lecteur qui aurait oublié le rôle des prépositions, adverbes, articles et autres conjonctions peut consulter l'annexe A.

2.2.1 Ambiguïtés

Le problème qui se pose est que plusieurs catégories peuvent très souvent convenir à une même forme. En Français, les formes ambiguës sont estimées à environ 25 % du lexique, voire plus pour les mots les plus courants [32]. Nous illustrons ce problème à la figure 1 en montrant toutes les combinaisons de catégories possibles pour une simple phrase.

2.2.2 Etiquetage automatique

étiquetage morpho-syntaxique

L'étiquetage automatique des mots par leurs catégories est un problème bien connu et assez bien maîtrisé pour l'écrit des langues les plus étudiées. Il consiste à gérer les

[1] Des catégories très spécifiques permettent une analyse syntaxique plus précise, mais sont plus difficiles à attribuer sans ambiguïté. Ainsi il serait intéressant de savoir si un verbe est transitif* ou non, mais cette information nécessite des connaissances lexicales importantes.

[2] La liste complète est disponible en annexe A.2.

ambiguïtés décrites ci-dessus et à attribuer (au besoin avec un taux de probabilité) une seule catégorie à chaque mot du texte. Les techniques employées pour parvenir à ce but sont diverses, mais ont toutes pour principe commun l'utilisation du contexte de chaque mot, c'est-à-dire l'élimination des combinaisons impossibles ou improbables. Ainsi, pour les trois derniers mots de l'exemple de la figure 1, il est hautement improbable que la combinaison "préposition + article + verbe" soit la bonne, car elle ne correspond à aucun schéma de la langue.

Les méthodes les plus utilisées pour parvenir à ce type de résultat sont :

- *Les règles linguistiques*, utilisées notamment par Intex [91] pour le Français et EngCG-2 [83] pour l'Anglais. Il s'agit en quelque sorte d'un analyseur syntaxique. On met en place des règles de grammaire, et seules les combinaisons permettant de respecter ces règles sont retenues. Ces méthodes sont lourdes à mettre en œuvre mais permettent d'élargir le contexte sans limitation.
- *Les modèles probabilistes*, souvent basés sur les chaînes de Markov cachées. Ils utilisent des corpus de textes manuellement étiquetés [22, 31, 89] ou non étiquetés [26]. A partir de ces données d'entraînement, ils calculent la probabilité pour qu'une catégorie apparaisse, connaissant les catégories des mots précédents. Leur avantage est qu'ils sont indépendants de la langue traitée. Parmi eux, TreeTagger [89], disponible gratuitement pour l'Anglais et le Français (entre autres) est celui que nous avons utilisé tout au long de nos travaux.
- *L'étiquetage par transformation* (*"Transformation-based learning"*). Mise en œuvre en particulier par Brill [18], cette technique se base également sur un apprentissage de règles grâce à un corpus préalablement étiqueté. Elle met en place des niveaux de règles de transformation de plus en plus précises visant à corriger les imprécisions des niveaux précédents. Ainsi, dans notre exemple, le *"l' "* aurait d'abord reçu l'étiquette "article" (cas le plus fréquent). Puis une règle aurait précisé : "Transformer un article en pronom s'il est suivi d'un verbe", conduisant à l'étiquette correcte.

Les méthodes récentes annoncent des précisions supérieures à 95 % pour les langues européennes. Plus de détails sur les différents algorithmes peuvent être trouvés dans différents supports [79, chap. 5][56, chap. 8][65, chap. 10].

2.3 Les autres informations morphosyntaxiques

La plupart des étiqueteurs automatiques fournissent comme seul renseignement la catégorie grammaticale. Il existe pourtant d'autres informations relatives à la morphologie.

lemme

Lemme. Le lemme est la racine d'un mot, dépouillée des marques d'accord, de conjugaison, de cas. C'est la forme qui se trouve en général en entrée du dictionnaire : en Français, le verbe à l'infinitif, le substantif au masculin singulier. Toute analyse de type sémantique nécessite la connaissance de ce lemme.

flexion

Flexions. Les flexions sont les modifications opérées sur le lemme pour distinguer les formes de conjugaison (personne, temps, mode, voix – flexion verbale) ou le genre, le nombre et le cas (flexion nominale)[1]. Ces aspects sont illustrés à la figure 2. L'opération

[1] Les langues utilisant les flexions sont dites *flexionnelles*.A l'opposé, dans les langues *isolantes*, comme le Chinois, toutes ces marques sont exprimées par des morphèmes distincts et séparés du lemme. Les langues *agglutinantes*, elles, à l'exemple du Japonais, comportent des affixes aux rôles flexionnels bien précis et clairement analysables.

qui consiste à retrouver ces informations est la *lemmatisation*[1]. En Anglais, les flexions sont peu nombreuses, et il est aisé de les retrouver à partir du lemme et de la catégorie grammaticale. En Français, la tâche est plus ardue. Le logiciel *Flemm* [74] permet de retrouver les flexions d'un texte écrit en Français à partir du résultat des analyseurs morphosyntaxiques Brill ou TreeTagger.

lemmatisation

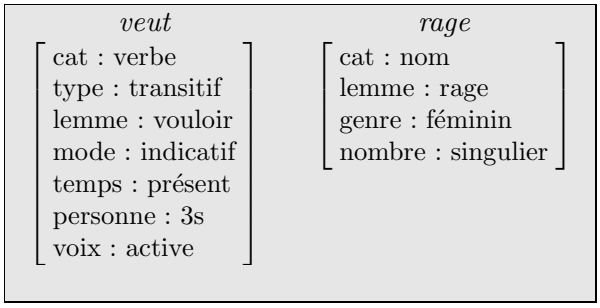


FIG. 2 – Description morphosyntaxique d'un verbe et d'un nom (voir figure 1).

3 Syntaxe

Un enfant parle très bien sa langue maternelle et pourtant il ne saurait en écrire la grammaire. [...] Le grammairien est celui qui sait pourquoi et comment l'enfant connaît la langue.
Umberto Eco, *Apostille au Nom de la Rose.*

A l'issue d'une analyse morphosyntaxique, les formes initiales présentes dans un énoncé sont remplacées par une liste ordonnée d'éléments contenant un certain nombre d'informations, parmi lesquelles la catégorie grammaticale et, éventuellement, le lemme, les flexions ou d'autres connaissances dont la présence dépend de l'application souhaitée.

La syntaxe décrit comment ces éléments s'ordonnent pour créer des *constituants*, composant aux-mêmes des *phrases*. De ce fait, on représente souvent le résultat d'une analyse syntaxique de façon hiérarchique.

syntaxe

A l'heure actuelle, les analyseurs syntaxiques de la langue naturelle sont loin d'afficher d'aussi bonnes performances que les analyseurs morphosyntaxiques. Pour parvenir à de bons résultats, il est nécessaire d'adapter les techniques globales à des problèmes spécifiques (langage restreint, type de textes particulier, finalité limitée...). Par exemple, une grammaire conçue dans un but de "compréhension" de phrases générales se devra d'être *robuste* (d'accepter le plus de tournures possibles) mais pourra se permettre d'accepter des constructions totalement agrammaticales qui ne peuvent se rencontrer en pratique. En revanche un système de génération de texte doit éviter toute forme incorrecte, mais peut se dispenser de certaines tournures trop complexes.

3.1 Du mot à la phrase

3.1.1 Constituants

En linguistique structurale, si l'on examine la construction d'une phrase du bas (les mots) vers le haut (la phrase), les étapes intermédiaires constituent la formation de *constituants*, ou *syntagmes*. Ces syntagmes sont qualifiés par le type de l'élément

syntagme

[1] Ce terme de lemmatisation est souvent employé pour signifier la racinisation (ou *stemming*) que nous décrivons à la section 5.1. Il est pourtant important de distinguer les deux. En particulier, la racinisation n'aboutit pas obligatoirement à des formes existant dans le lexique.

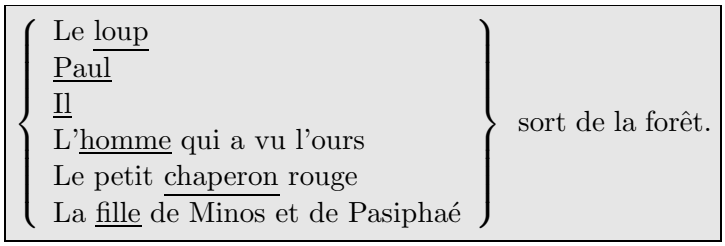


FIG. 3 – Substituabilité du syntagme (ici, un syntagme nominal, avec à sa tête l'élément souligné – un nom ou un pronom qui remplace lui-même un nom). Notons que cette caractéristique est vraie pour des genres, nombres et cas identiques. Le sujet de la phrase ne pourrait pas être remplacé par *"les loups"* ou *"lui"*, qui sont pourtant également des syntagmes nominaux.

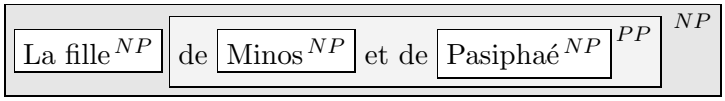


FIG. 4 – Imbrication de syntagmes, avec *NP* = Noun Phrase (syntagme nominal) ; *PP* = Prepositional Phrase (groupe prépositionel).

tête

principal (la *tête*) : syntagmes nominaux, verbaux, adjectivaux, prépositionnels, adverbiaux. Les éléments autres que la tête sont des *spécifieurs* (déterminants...), des *qualificateurs* (adjectifs, adverbes...), des *compléments* (compléments du nom, propositions relatives...).

Il est important de remarquer que ce qui forme un constituant est dépendant du contexte [82], comme l'illustre l'exemple suivant, dans lequel *"François et Ségolène"* forme tantôt un syntagme à part entière, tantôt des éléments séparés :

(5) [*François et Ségolène*] forment un ménage moderne.

(6) Les enfants sont gardés par [*François*] *et* [*Ségolène*] part en voyage.

Ainsi rechercher des types de syntagmes à un niveau purement local sans se préoccuper des autres types de construction possibles peut conduire à des erreurs.

Par ailleurs, les syntagmes de même type (nominal, verbal, etc.) ont la même fonction dans la phrase que leur tête, et ils sont *syntaxiquement* substituables entre eux (comme le montre la figure 3). Les syntagmes peuvent s'imbriquer les uns dans les autres (figure 4[1]), jusqu'à former des *propositions*, unité supérieure constituée d'un sujet, d'un verbe et d'éventuels compléments.

3.1.2 Phrases

Au niveau syntaxique, la phrase est l'unité prépondérante de l'analyse. Elle peut être constituée d'une seule proposition mais admet également la coordination (8) et la subordination (9) des propositions :

[1] Les abréviations des constituants que nous donnons ici et que nous utiliserons tout au long du mémoire sont les abréviations anglaises, universellement utilisées. Par exemple : *VP* pour *Verbal Phrase* (syntagme verbal), *S* pour *Sentence* (phrase), etc. (voir l'annexe B.5).

(7) [Deux pigeons s'aimaient d'amour tendre].[1]

(8) [Deux sûretés valent mieux qu'une], et [le trop en cela ne fut jamais perdu].[1]

(9) [Vous savez que [nul n'est prophète en son pays]].[1]

Sa définition est surtout intuitive : il s'agit d'une suite de mots considérée comme complète du point de vue du "sens". A l'écrit, une phrase commence par une majuscule et se termine par une ponctuation forte, mais nous avons vu que cela ne constitue pas des conditions suffisantes. Une succession de phrases forment le *discours*.

3.1.3 Les fonctions grammaticales

fonction grammaticale

La fonction grammaticale est le rôle *syntaxique* qu'occupe un composant par rapport à un autre constituant. Elle dépend surtout de la position physique des constituants. Par exemple, sauf exception (voix passive, interrogation, etc.), le *sujet* d'un verbe est, en Français comme en Anglais, placé avant lui. Les autres fonctions essentielles sont l'objet et l'objet indirect, le complément du nom... Les relations basées sur une préposition (*"dans"*, phrase 13) peuvent prendre le nom de celle-ci (en l'absence d'apport sémantique). Notons également que pour la phrase 12, le sujet réel a pris la place du sujet grammatical (*"le chaperon"*) pour tenir compte de la voie passive.

(10) [Le loup] [lorgne] [le petit chaperon rouge].
sujet — *objet*

(11) [Le petit chaperon rouge] [dit] [à sa grand-mère] [qu'elle a de grandes mains].
sujet — *obj. indirect* — *objet*

(12) [Le chaperon] [n'est pas alerté] [par [les grandes dents] [de sa grand-mère]].
objet — *sujet* — *complément du nom*

(13) [Le chaperon] [finira] [***dans*** le ventre du loup] !
sujet — *dans*

Il ne faut pas confondre les relations grammaticales avec les relations sémantiques (*agent*, *bénéficiaire*, *lieu*) qui seront abordées plus loin[2]. Ces deux types de relations sont liés, mais il n'existe pas de moyen unique de passer de l'une à l'autre.

3.2 Les ambiguïtés syntaxiques

One morning I shot an elephant in my pajamas. How he got into my pajamas I don't know.
Groucho Marx, *Animal Crackers*.

"Parfaitement, dit le Major. Ma grand-mère, qui est morte, y avait un appartement et mon père l'a conservé."
Le Bison n'entendant pas d'e muet à la fin comprit qu'il s'agissait de l'appartement et non de la grand-mère.
Boris Vian, *Les Remparts du Sud*.

[1] *Jean de la Fontaine.*
[2] Section 4.2.2 page 25.

Les ambiguïtés concernant la syntaxe d'un énoncé peuvent avoir de nombreuses causes. Celles-ci ont toutes pour résultat une déduction erronée de l'agencement des mots dans la phrase. On peut classer les ambiguïtés syntaxiques selon les connaissances nécessaires pour les résoudre :

Connaissances pragmatiques du monde qui nous entoure ou, dans les exemples suivants, du contexte (nous avons besoin de connaître les occupations récentes de Jean, qui peut venir du Marché aux Puces ou de Pekin) :

(14) a. Jean [a rapporté ^{Verbe}] [un vase ^{NP}] [de Chine ^{PP}]

$NP \rightarrow NP\ PP$

b. Jean [a rapporté ^{Verbe}] [un vase ^{NP}] [de Chine ^{PP}]

$VP \rightarrow VP\ NP$

Connaissances sémantiques ajoutées par exemple au lexique (ici, une tarte a des ingrédients et les pommes sont comestibles, ce qui n'est pas le cas des clients, auxquels on vend des choses) :

(15) a. Jean [vend ^{Verbe}] [une tarte ^{NP}] [aux pommes ^{PP}]

$NP \rightarrow NP\ PP$

b. Jean [vend ^{Verbe}] [une tarte ^{NP}] [aux clients ^{PP}]

$VP \rightarrow VP\ PP$

Connaissances syntaxiques comme par exemple les accords en genre et en nombre :

(16) a. un [jus ^{Nom}] d'[oranges ^{Nom}] [fraîches ^{Adj}]

Nom + Adj

b. un [jus ^{Nom}] d'[oranges ^{Nom}] [frais ^{Adj}]

Nom + Adj

(17) a. des [tags ^{Nom}] dans les [banlieues ^{Nom}] [karchérisés ^{Adj}]

Nom + Adj

b. des [tags ^{Nom}] dans les [banlieues ^{Nom}] [karchérisées ^{Adj}]

Nom + Adj

L'exemple suivant rend peut-être encore mieux compte de l'ampleur du problème : le groupe *"à la rhubarbe"*, qui s'attache bien entendu à *"la tarte"*, peut être lié au verbe *"manger"* pour deux raisons différentes : en tant que complément d'object indirect (comme dans *"il a donné la tarte à sa belle-mère"*), si l'on ignore que *"manger"* n'a pas d'emploi transitif indirect, ou en tant qu'adverbial (comme dans *"il a mangé la tarte à la maison"*).

(18) Il [a mangé] [la tarte] [à la rhubarbe]

complément

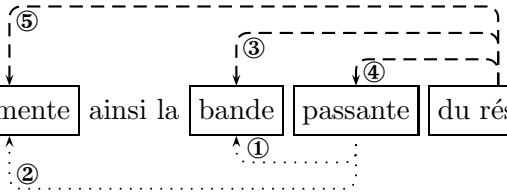
objet indirect

adverbial

En l'absence des connaissances pragmatiques, sémantiques et/ou syntaxiques nécessaires (et aucun système ne les possède toutes), une analyse pourra produire plusieurs résultats pour un ensemble donné de règles. Ainsi, *Froissart* [39] montre que pour la simple phrase *"On augmente ainsi la bande passante du réseau"*, pas moins de 11 solutions sont engendrées par un analyseur classique. Parmi les ambiguïtés soulignées, on peut citer (exemple 19) :

- l'adjectif *"passante"* peut être accolé :
  ① au nom *"bande"* (comme cela doit être compris) ;
  ② au verbe *"augmente"*, comme dans *"On [rend ainsi la bande] [passante]"*.
- *"de réseau"* peut être complément :
  ③ du nom *"bande"* (correct) ;
  ④ de l'adjectif *"passante"*, comme dans *"[traversée] [de part en part]"*;
  ⑤ du verbe *"augmente"*, comme dans *"on [augmente] les salaires [de la moitié]"*.

(19) On augmente ainsi la bande passante du réseau.



Les cas les plus fréquents d'ambiguïtés syntaxiques sont liés au rattachement des groupes prépositionnels et des propositions relatives. Ils peuvent très vite mener à des centaines de constructions possibles, même avec des phrases relativement courtes.

A titre d'expérience, nous avons soumis la phrase suivante : *"We are searching sections dealing with version management in articles containing a section about object databases."*[1] à deux analyseurs gratuits disponibles en ligne, LinGO ERG [24][2] et Link Grammar [92][3], parvenant à respectivement 159 et 543 analyses possibles.

L'annexe C détaille deux méthodes pour réduire les ambiguïtés syntaxiques en l'absence de connaissances sémantiques.

3.3 L'analyse syntaxique

3.3.1 L'analyse superficielle

analyse superficielle

L'analyse superficielle (ou partielle) [2] a pour but de reconnaître les syntagmes simples, non récursifs, d'un énoncé, sans les lier les uns aux autres. Ainsi, les principaux attachements responsables des ambiguïtés citées ci-dessus, comme les attachements prépositionnels, ne sont pas traités (du moins dans un premier temps). L'idée est d'obtenir des résultats certes moins riches, mais plus rapides et plus sûrs. Elle est appropriée pour un certain nombre d'applications qui ne cherchent pas à définir des dépendances précises, comme par exemple la reconnaissance de syntagmes nominaux [17]. Cette approche s'oppose à l'analyse profonde (ou complète), qui cherche à regrouper chaque phrase dans une unique représentation.

analyse profonde

[1] "Nous cherchons des sections traitant de la gestion des versions dans un article contenant une section sur les bases de données objet." Cette phrase est inspirée d'une requête d'INEX 2004 (Topic. 130), avec uniquement des modifications de vocabulaire, car certains mots n'étaient pas reconnus par les analyseurs utilisés.

[2] http://www.delph-in.net/erg/

[3] http://hyper.link.cs.cmu.edu/link/index.html

```
:niveau0
    verbe      →  aux past_part
    np         →  art nom
:niveau1
    pp         →  prep np
    vp         →  verbe np
:niveau2
    prop_rel   →  pro_rel vp
```

FIG. 5 – Cascade de règles d'automates à états finis.

Voici un exemple d'analyse partielle, dans laquelle on regroupe les syntagmes nominaux, verbaux et prépositionnels non récursifs. On peut également reconnaître une clause C, non ambiguë, mais dans laquelle le PP n'est pas inclu.

(20) [Les gendarmes ^{NP}] [interpellent ^{VP}] [un conducteur ^{NP}] ^{C} [en état d'ivresse ^{PP}]

Cass

Un exemple d'analyseur superficiel efficace est Cass [3], qui consiste en une cascade d'automates à états finis. Chacun de ces automates représente un niveau de reconnaissance. Au niveau le plus bas, le niveau 0, l'entrée est une suite de mots avec leur catégorie grammaticale (sortie d'un analyseur morphosyntaxique comme TreeTagger). L'automate de niveau 1 trouve toutes les séquences du niveau 0 correspondant à un certain motif – par exemple, des syntagmes nominaux – et les réduit à des éléments simples portant un nouveau nom – par exemple, `np`. La sortie de cet automate devient l'entrée de l'automate suivant, et ainsi de suite.

La figure 5 propose une cascade de trois ensembles de règles simples, et la figure 6 illustre l'application de ces règles à la phrase suivante :

(21) Le dessin d'un boa qui a avalé un éléphant.

Notons que les éléments utilisés dans les règles, s'ils participent à la création d'un nouvel élément, ne sont plus accessibles au niveau suivant (par exemple, une référence à un `nom` au niveau 1 serait inopérante). Remarquons également qu'un seul résultat est fourni, les ambiguïtés n'étant pas prises en compte (le premier résultat trouvé, l'analyse est interrompue).

3.3.2 Les grammaires hors-contexte

grammaire hors-contexte

Une grammaire hors-contexte (ou CFG, pour *context-free grammar*) se compose d'un ensemble de règles de la forme :

$$E \rightarrow E_1 \ \ldots \ E_n$$

qui expriment le fait que la séquence d'expressions $\{E_1 \ \ldots \ E_n\}$ peut être remplacée (*réécrite*) par un nouvel identifiant *unique* E, en faisant abstraction des éléments qui l'entourent. L'application de ces règles, illustrées à la figure 7 par des regroupements de *constituants* linguistiques, est souvent représentée de façon arborescente, comme le montre la figure 8.

Les CFG permettent de refléter les aspects hiérarchiques des constructions grammaticales par l'imbrication des constituants et la *récursivité* des règles.

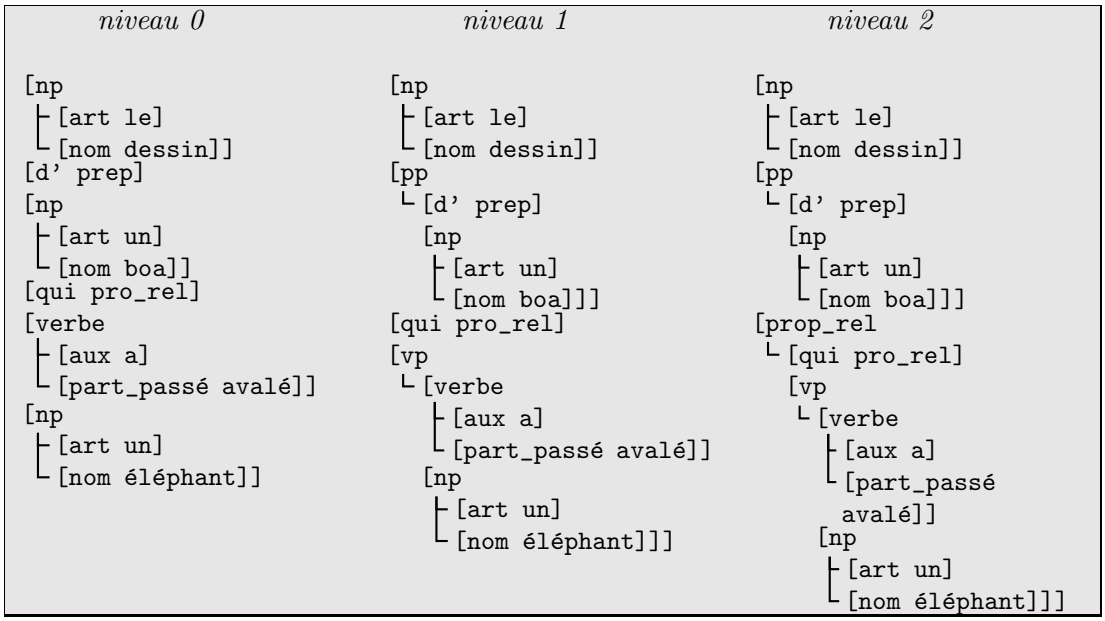


FIG. 6 – Analyse superficielle en cascade.

Règles		*Exemples*
S	$\rightarrow$ *NP VP*	[Le loup]$_{NP}$ [sort de la forêt]$_{VP}$
NP	$\rightarrow$ *Pronom*	il
NP	$\rightarrow$ *Nom_Propre*	Paul
NP	$\rightarrow$ *Det Adj? Nom Adj?*	[Le]$_{DET}$ [petit]$_{ADJ}$ [chaperon]$_{NOM}$ [rouge]$_{ADJ}$
NP	$\rightarrow$ *NP PP*	[La fille]$_{NP}$ [de Minos et de Pasiphaé]$_{PP}$
VP	$\rightarrow$ *Verbe PP*	[sort]$_{Verbe}$ [de la forêt]$_{PP}$
VP	$\rightarrow$ *Verbe NP*	[mange]$_{Verbe}$ [le chat]$_{NP}$
PP	$\rightarrow$ *Prep NP*	[de]$_{Prep}$ [la forêt]$_{NP}$
...	$\rightarrow$ ...	...

FIG. 7 – Règles hors-contexte permettant d'obtenir les constructions vues à la figure 3. Avec S : *sentence* (phrase) ; *NP* : *noun phrase* (syntagme nominal) ; *VP* : *verbal phrase* (syntagme verbal) ; *PP* : *prepositional phrase* (syntagme prépositionnel) ; *Det* : déterminant ; *Adj* : adjectif ; *Prep* : préposition. Le point d'interrogation '*?*' signifie que l'élément est optionnel.

Comme nous le verrons dans les sections suivantes, de telles grammaires provoquent vite une combinatoire élevée. Par ailleurs, utilisées telles quelles, elles ne fournissent que peu d'informations sur la structure de la phrase, puisqu'elles ne donnent pas la nature des relations qui lient les constituants les uns aux autres.

Plusieurs stratégies d'analyse existent pour traiter des grammaires hors-contexte, de façon descendante ou ascendante d'une part, en profondeur ou en largeur d'autre part. La description, les avantages et les inconvénients de chaque possibilité sont décrits par exemple par *Gardent et Baschung* [43, chap. 2].

3.3.3 L'analyse en dépendance

analyse en dépendance

L'analyse syntaxique en dépendance diffère surtout de l'analyse en constituants (comme les règles hors-contexte) par le mode de représentation. Les deux approches n'ont pas de différence au niveau de leur couverture ou de leur expressivité.

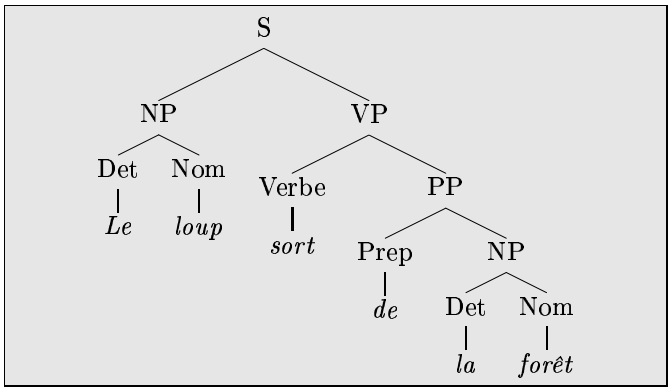


FIG. 8 – Arbre syntaxique de la phrase *"Le loup sort de la forêt"*, représentant l'application de certaines règles proposées à la figure 7.

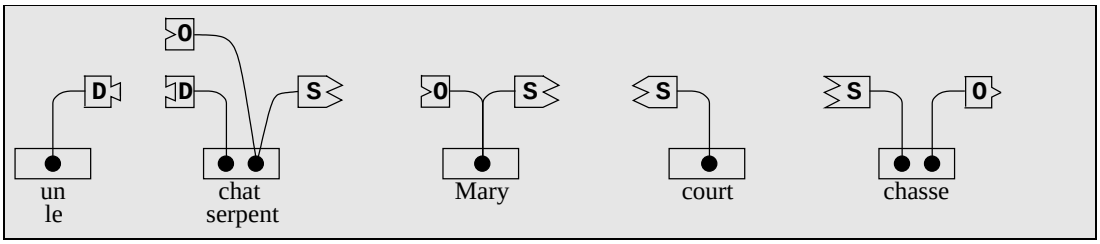


FIG. 9 – Définition du lexique de la *Link Grammar*.

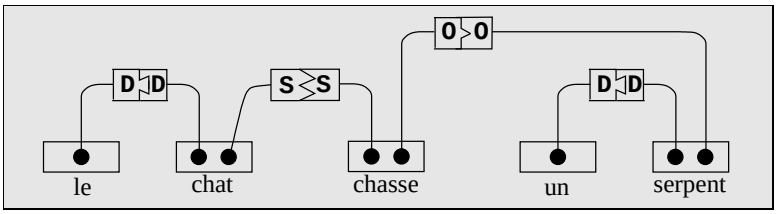


FIG. 10 – Analyse syntaxique avec la *Link Grammar*.

L'idée est de relier des mots, et non plus des constituants, en définissant pour chaque catégorie une liste (avec contraintes) d'autres catégories susceptibles de lui être attachées. Puis l'analyseur se charge de mettre en place les liens.

Un exemple de grammaire de dépendance est la *Link Grammar* [92]. Son fonctionnement est rapidement illustré aux figures 9 (définition du lexique et des contraintes d'attachement) et 10 (application). Notons que les liens ne doivent pas se croiser.

3.3.4 Les traits

Les grammaires décrites jusqu'à présent ne suffisent absolument pas à capturer de façon utile le langage naturel. Le première raison est qu'il existe de nombreuses restrictions lors du regroupement des mots en constituants. Par exemple, *"*le fille"* ou *"*les loup"* ne sont pas corrects car les accords en genre et en nombre ne sont pas respectés. La seconde raison est que l'on ne souhaite pas seulement reconnaître des constructions, mais aussi en extraire des informations linguistiques, permettant d'effectuer une analyse adaptée à l'application mise en œuvre.

traits

Ces deux problèmes peuvent trouver leur réponse avec l'ajout de *traits*, des paires

```
s --> np, vp.                nom --> [forêt].
np --> det, nom.             verbe --> [sort].
vp --> verbe, pp.            prep --> [de].
pp --> prep, np.             det --> [le].
nom --> [loup].              det --> [la].
```

FIG. 11 – Exemple de règles de grammaire à clauses définies.

d'attributs/valeurs attachées aux entrées lexicales et/ou aux constituants formés durant l'analyse. Par exemple, ajouter à certaines entrées du lexique (déterminants, noms, verbes) des traits "genre" et "nombre", prenant respectivement les valeurs *masculin* ou *féminin* d'une part, *singulier* ou *pluriel* d'autre part, permet de proposer, la règle hors-contexte augmentée suivante :

$$NP \rightarrow \quad Det\ Nom$$
si Det et Nom s'accordent en genre et en nombre

soit

$$\underset{\begin{bmatrix}\text{genre} : \text{G}\\ \text{nombre} : \text{N}\end{bmatrix}}{NP} \rightarrow \underset{\begin{bmatrix}\text{genre} : \text{G}\\ \text{nombre} : \text{N}\end{bmatrix}}{Det} \quad \underset{\begin{bmatrix}\text{genre} : \text{G}\\ \text{nombre} : \text{N}\end{bmatrix}}{Nom}$$

Ainsi, seuls les syntagmes nominaux accordés sont acceptés par la grammaire, et l'information du genre et du nombre est propagée pour la suite de l'analyse.

Un autre exemple intéressant est la *forme* et la *sous-catégorisation* des verbes. Ainsi l'information qu'un verbe est à l'impératif permettra d'empêcher qu'on lui attache un sujet. La connaissance de la valence d'un verbe (transitivité* ou pas) évitera certaines ambiguïtés[1].

Les traits peuvent être utilisés à tous les niveaux (phonologique, morphologique, syntaxique, sémantique, pragmatique), et il est possible d'ajouter toutes sortes de traits nécessaires à une application donnée.

3.3.5 Les grammaires à clauses définies

grammaire à clauses définies

Les grammaires à clauses définies (ou DCG, pour *Definite Clause Grammars*) sont des programmes écrits dans le langage Prolog[2] qui implémentent directement les grammaires hors-contexte, de la façon simple indiquée par le *code* de la figure 11. La stratégie "imposée" par le fonctionnement de Prolog est l'analyse descendante en profondeur, de gauche à droite.

Les DCG ont l'avantage de permettre l'extension des grammaires, par l'ajout d'informations syntaxiques, sémantiques ou autres, véhiculées par chaque constituant. Ainsi, si l'on veut conserver le texte analysé au fur et à mesure, on peut utiliser des prédicats à la place des atomes de la figure 11, et propager l'information dans l'arbre. C'est ce qu'illustre la figure 12. Précisons qu'en Prolog, les variables commencent par une majuscule, les constantes et les prédicats par une minuscule. Les crochets `'['` et `']'`

[1] Par exemple, *"clients"* peut être attaché à *"vend"* dans *"Marie vend une tarte aux clients"*, car *"vendre"* est transitif indirect*, mais *"pommes"* sera sans ambiguïté attaché à *"tarte"* dans *"Marie cuisine une tarte aux pommes"*, car *"cuisine"* n'admet pas d'objet indirect.

[2] Prolog est un langage de programmation logique dont un aperçu rapide et clair est donné par *Blackburn et Bos* [15, annexe D]. Les bases essentielles à la compréhension seront précisées au fur et à mesure du texte.

```
s(Words) --> np(NP), vp(VP), {append(NP, VP, Words)}.
np([Det, Nom]) --> det(Det), nom(Nom).
vp([Verbe | PP]) --> verbe(Verbe), pp(PP).
pp([Prep | NP]) --> prep(Prep), np(NP).
nom(loup) --> [loup].
nom(forêt) --> [forêt].
verbe(sort) --> [sort].
prep(de) --> [de].
det(le) --> [le].
det(la) --> [la].
```

FIG. 12 – Exemple de règles étendues de grammaire à clauses définies.

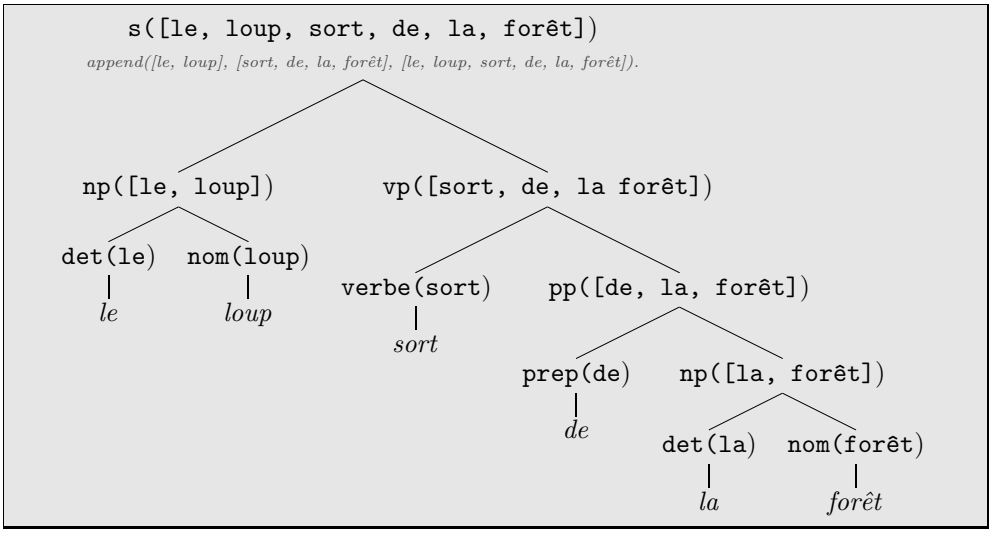


FIG. 13 – Propagation des informations par la DCG de la figure 12.

représentent une liste, et `[E | Rest]` est une liste commençant par l'élément `E` et continuant par les éléments de la liste `Rest`. Enfin, le prédicat `append(A, B, C)` permet de concaténer la liste `A` et la liste `B` pour obtenir la liste `C`. Dans la syntaxe des grammaires à clauses définies, les instructions Prolog spécifiques sont insérées entre accolades (voir le prédicat `s`).

L'arbre de la figure 13 montre l'application de la grammaire, avec la propagation des éléments demandés. Il est ainsi possible de dépasser le stade de la reconnaissance simple de constructions et, en utilisant les traits* jugés nécessaires, de transmettre tout type d'indications pendant l'analyse. Celles-ci peuvent servir à édifier une interprétation sémantique, des relations grammaticales ou toute autre représentation utile, mais aussi à opérer une analyse plus fine des énoncés.

Ainsi, on peut modéliser les contraintes de genre et de nombre, comme le fait la nouvelle grammaire de la figure 14, qui permet de gérer certaines classes d'ambiguïtés syntaxiques décrites à la section 3.2 (exemples 16 et 17, page 12). Le caractère `'_'` indique que la variable n'est pas définie ou pas utilisée. Ce nouvel ensemble de règles a deux avantages :

- Il rejette les syntagmes incorrects comme *"*une jus"*, car le déterminant et le nom ne partagent pas le même nombre.
- Il attache *sans ambiguïté* les adjectifs *"fraîches"* aux *"oranges"* dans *"un jus d'oranges fraîches"*, et *"frais"* au *"jus"* dans *"un jus d'oranges frais"*. En effet,

```
np(Nombre, Genre) --> nom(Nombre, Genre), adj(Nombre, Genre).
np(Nombre, Genre) --> nom(Nombre, Genre), prep, np(_, _).
np(Nombre, Genre) --> det(Nombre, Genre), nom(Nombre, Genre).
np(Nombre, Genre) --> nom(Nombre, Genre).
nom(sing, masc) --> [jus].
nom(plur, fem) --> [oranges].
adj(sing, masc) --> [frais].
adj(plur, fem) --> [fraîches].
det(sing, masc) --> [un].
det(sing, fem) --> [une].
prep --> [d'].
```

FIG. 14 – Exemple de règles étendues de grammaire à clauses définies.

```
s(Sem) --> np(NP), vp(VP), {append(NP, VP, Sem)}.
np([Nom]) --> det, nom(Nom).
vp([Verbe | PP]) --> verbe(Verbe), pp(PP).
pp(NP) --> prep, np(NP).
nom(animal:loup) --> [loup].
nom(lieu:forêt) --> [forêt].
verbe(mouvement:sortir) --> [sort].
prep --> [de].
det --> [le].
det --> [la].
```

FIG. 15 – Exemple de règles étendues de grammaire à clauses définies.

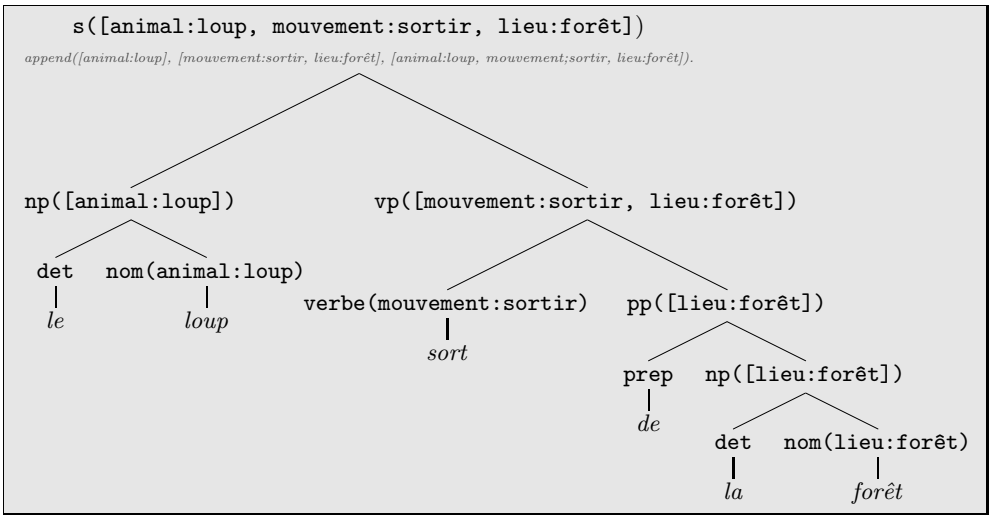


FIG. 16 – Propagation des informations par la DCG de la figure 15.

pour le premier NP, le syntagme *"un jus d'oranges"*, dont la tête* est *"jus"*, est au masculin singulier, tandis que *"fraîches"* est au féminin pluriel. L'attachement de *"fraîches"* à *"jus"* est donc impossible. Le second NP est traité de la même façon.

Un dernier exemple permet de faire quelques pas dans la sémantique, avec l'enrichissement de la figure 11 par des prédicats contenant des informations lexicales sémantiques, comme l'illustre la figure 15. L'application de ces règles est illustrée à la figure 16.

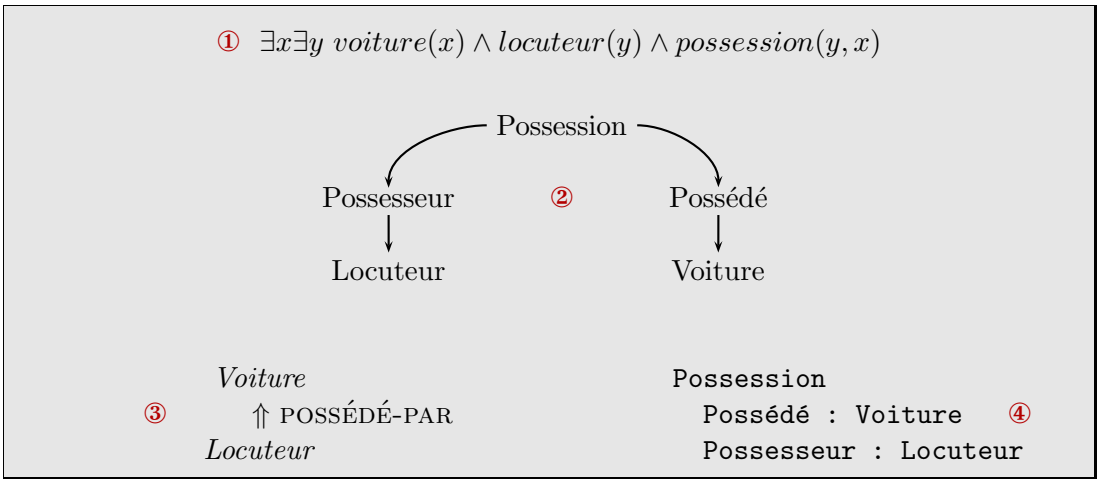


FIG. 17 – Différentes représentations sémantiques de la phrase *"J'ai une voiture"* (d'après *Jurafsky et Martin* [56, chap. 14]).

3.3.6 Les formalismes plus évolués

De nombreux formalismes de grammaire bien plus évolués, mais aussi bien plus complexes, que ceux que nous avons évoqués ici, ont été développés durant les dernières décennies. Les grammaires *à traits* comme GPSG (Generalized Phrase Structure Grammar [1, chap. 2]) ou LFG (Lexical Functional Grammar [1, chap. 1]) ou les grammaires d'*unification* comme FUG (Functional Unification Grammar [60]) ou surtout HPSG (Head-driven Phrase Structure Grammar [1, chap. 3][14]), sont des théories linguistiques à part entière, articulant souvent le lexique, la syntaxe et la sémantique dans une même représentation.

4 Sémantique

Les applications des analyses sémantiques sont diverses et plus ou moins ambitieuses. Un apport sémantique peut servir à réduire les ambiguïtés syntaxiques, à mieux cibler des concepts (par exemple en recherche d'information), mais sa finalité globale est de représenter formellement[1] l'information véhiculée par un énoncé et éventuellement d'en inférer de nouvelles connaissances ou une réponse à la question posée (si l'énoncé est une question).

Nous abordons tout d'abord la sémantique *grammaticale*, qui s'attache à construire le sens de l'énoncé global, puis à la sémantique *lexicale*, qui étudie la contribution des mots à ce sens et toutes les ambiguïtés qu'ils provoquent.

4.1 La représentation du sens

Nous décrivons ici comment il est possible de symboliser le sens d'un énoncé par une représentation *logique*, à l'aide de prédicats.

D'autres formalismes ont été proposés (voir de courts exemples à la figure 17), que nous ne traiterons pas, notamment les *réseaux sémantiques* ②, les *dépendances*

conceptuelles ③ et les représentations par cadres (*frames* ④).

4.1.1 La représentation logique

La formule logique (exemple ① de la figure 17), grâce à un langage formel connu, possédant une syntaxe simple et dénuée d'ambiguïté, offre la possibilité :

- de représenter un énoncé :

  (22) Je ne te hais point.
  $\exists x \exists y\ locuteur(x) \wedge auditeur(y) \wedge \neg haïr(x, y)$

- d'attribuer des *valeurs de vérité* binaires à une expression donnée :

  (23) $empereur(Napoléon)$ est vrai ssi Napoléon est empereur.

- de raisonner sur des connaissances et des énoncés :

  (24) si $\forall x(homme(x) \Rightarrow mortel(x))$
  et $homme(Socrate)$
  alors $mortel(Socrate)$

Mais la logique des prédicats du premier ordre est loin de pouvoir traiter tous les phénomènes des énoncés en langage naturel, dont le sens ne se limitent pas à leur seule valeur de vérité. En particulier :

- La logique gère sans état d'âme certaines subtilités du langage :

  (25) Je suis en Espagne. $\rightsquigarrow \exists x\ locuteur(x) \wedge est\_situé(x, Espagne)$

  (26) C'est en Espagne que je suis. $\rightsquigarrow \exists x\ locuteur(x) \wedge est\_situé(x, Espagne)$

- Seules les *propositions* auxquelles on peut attribuer aisément une valeur de vérité sont représentables. Au-délà, pour certains aspects du langage naturel, la formalisation devient très complexe, voire impossible, sans recourir à d'autres types de logiques (logique floue, logique modale, concept de *proposition exprimée*...). Par exemple :
  - la modalité (*"Il est possible/probable/nécessaire que..."*) ;
  - le temps ($empereur(Napoléon)$ n'est vrai qu'à un moment donné de l'histoire) ;
  - les souhaits (*"J'espère que vous viendrez"*)
  - les concepts flous (*"Paul est jeune"*, *"Beaucoup pensent que..."*;
  - l'impératif (*"Fais ce que je te dis !"*)
- Cette formalisation ne distingue pas la valeur linguistique de la valeur logique, et ne rejette pas les énoncés vrais mais absurbes (*Les Pyramides sont en Egypte et cet immeuble a 5 étages.*[1]).

Ces différents phénomènes n'ayant que peu de répercutions sur le travail que nous décrivons dans la seconde partie de ce mémoire, nous ne nous attarderons pas sur les moyens de les prendre en compte.

4.1.2 Un exemple de représentation, la DRT

Dans la Théorie de Représentation du Discours (ou DRT pour *Discourse Representation Theory* [57]), le sens des énoncés est représenté par des Structures de Représentation du Discours (DRS). Une DRS est une paire composée d'un ensemble de *référents de discours* et d'un ensemble de *contraintes*, ou de *conditions*, sur ces réfé-

DRT

DRS

référent

[1] Les mots sont importants, il s'agit bien entendu d'une *représentation* et non pas d'une *compréhension*, et son caractère *formel* s'oppose par essence au fait que les langues ne le sont pas, ou qu'au moins les auteurs ne les traitent pas de façon formelle [33].

[1] Même si la frontière entre une phrase absurde et une phrase qui fait du sens est mince et dépend beaucoup du contexte (*"J'habite à Saint-Etienne et je n'aime pas le football"*).

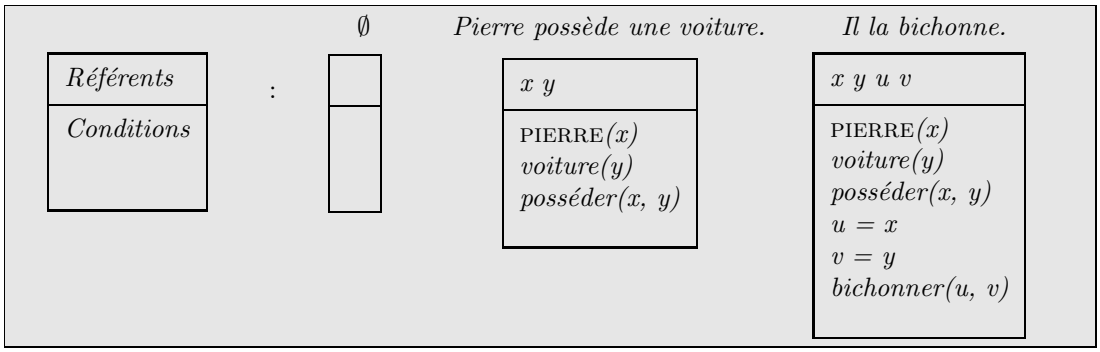


FIG. 18 – Exemple de structure de représentation du discours (DRS).

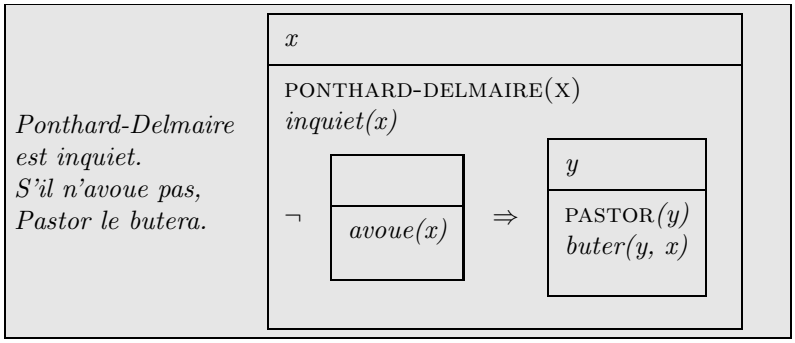


FIG. 19 – Exemple d'imbrication de DRS.

rents. Cette structure est généralement présentée à l'aide de "boîtes" à deux étages. Une DRS s'enrichit à chaque nouvel énoncé, comme le montre la figure 18.

Les *conditions* de la DRS peuvent contenir des prédicats, mais aussi d'autres DRS, éventuellement liées par des opérateurs $\neg$, $\vee$, $\Rightarrow$, comme en logique classique (voir figure 19). En revanche, les quantificateurs ne sont pas nécessaires, ils sont exprimés par les "boîtes".

Définie ainsi, une telle structure est équivalente à des expressions logiques traditionnelles. Pourtant, la DRT permet de gérer de façon élégante la négation, la portée des quantificateurs et les phénomènes de références comme les anaphores* ou les ellipses* (voir la section 4.3).

Chaque terme du lexique peut être associé à une représentation DRS de base, selon sa catégorie grammaticale. Ceci permet d'adapter la DRT à de nombreuses stratégies d'analyse sémantique (voir plus loin – section 4.2) [15, vol. II, chap. 2].

4.1.3 Les événements

rôle thématique

L'utilisation de prédicats uniques pour représenter les événements (et plus généralement les verbes) et les éléments qui y sont rattachés pose un certain nombre de problèmes théoriques et pratiques pour la détermination des rôles thématiques. Dans notre exemple 22, *"haïr"* étant un verbe transitif, un prédicat binaire (agent + thème) a été utilisé. Cette approche trouve très vite ses limites. Prenons l'exemple du verbe transitif *"manger"*, qui peut être utilisé dans les conditions suivantes [56, chap. 14] :

(27) a. J'ai mangé. *(emploi intransitif)*

b. J'ai mangé un sandwich.

c. J'ai mangé [un sandwich] [dans mon bureau].

d. J'ai mangé [un sandwich] [avec les doigts].

e. J'ai mangé [un sandwich] [pour le dîner].

f. J'ai mangé [un sandwich] [dans mon bureau] [pour le dîner].

g. ...

Utiliser un prédicat $manger(x, y)$ ne suffit bien sûr pas. Une autre possibilité serait d'utiliser un prédicat contenant l'ensemble des rôles possibles ($manger(v, w, x, y, z)$ pour le sujet, le thème (l'objet), le lieu, la manière, le moment), mais c'est à la fois trop riche (les variables resteront la plupart du temps non instanciées) et trop limitant (on peut manger *"avec les doigts et avec dégoût"* ou *"pour le dîner à 20h30"*, etc.).

La solution la plus réaliste est de représenter les événements comme des objets, au même titre que les noms, et de multiplier les prédicats. C'est le principe de la *réification*. L'action de manger introduit donc la clause $\exists m\ manger(m)$ ou $\exists m\ evt(m, manger)$, puis chaque nouveau syntagme se rattachant au verbe provoque l'insertion d'un nouveau prédicat :

réification

(28) J'ai mangé [un sandwich] [dans mon bureau] [pour le dîner].
$\exists m\ \exists s\ \exists b\ \exists d\ manger(m) \wedge sandwich(s) \wedge bureau(b) \wedge dîner(d)$
$\wedge\ agent(m, \text{SPEAKER}) \wedge thème(m, s) \wedge lieu(m, b) \wedge temps(m, d)$

4.2 L'analyse sémantique

L'analyse sémantique a pour but d'associer à une séquence de mots une représentation interne de son sens. Le formalisme choisi, le niveau de détails, la caractère complet ou partiel de l'analyse, les constructions reconnues, dépendent de l'utilisation que l'on souhaite faire du résultat et des connaissances actuelles en la matière.

Nous présentons ici quatre méthodes différentes d'analyse sémantique : l'analyse profonde, dont le but est d'obtenir une représentation logique complète de l'énoncé ; l'interprétation sémantique des relations grammaticales, qui s'appuie, comme la première, sur une analyse syntaxique totale ; les grammaires sémantiques, qui modélisent des domaines très spécifiques ; et les patrons sémantiques, qui détectent des informations prédéfinies dans un texte.

4.2.1 L'analyse profonde par compositionnalité

De nombreuses méthodes d'analyse s'inspirent fortement du *principe de compositionnalité* [53] et de la grammaire de Montague [78, 58], qui permettent d'associer l'élaboration du sens d'une phrase aux règles syntaxiques qui la régissent. Ainsi le sens d'un constituant est la composition des sens des éléments qu'il contient. Cette progression de bas en haut est illustrée par la figure 20 [1].

On voit que cette façon de procéder semble assez puissante, mais pose un certain nombre de difficultés pratiques.

La première réside dans la nécessité de faire correspondre les variables introduites par certaines catégories (dans notre exemple, x et y, par les articles) à des positions dans les prédicats générés par d'autres éléments (principalement les noms et les verbes – *manger(x, y)* ou *manger(y, x)* ?).

[1] Une illustration de la construction d'une DRS est également donnée pour l'exemple à la figure 21.

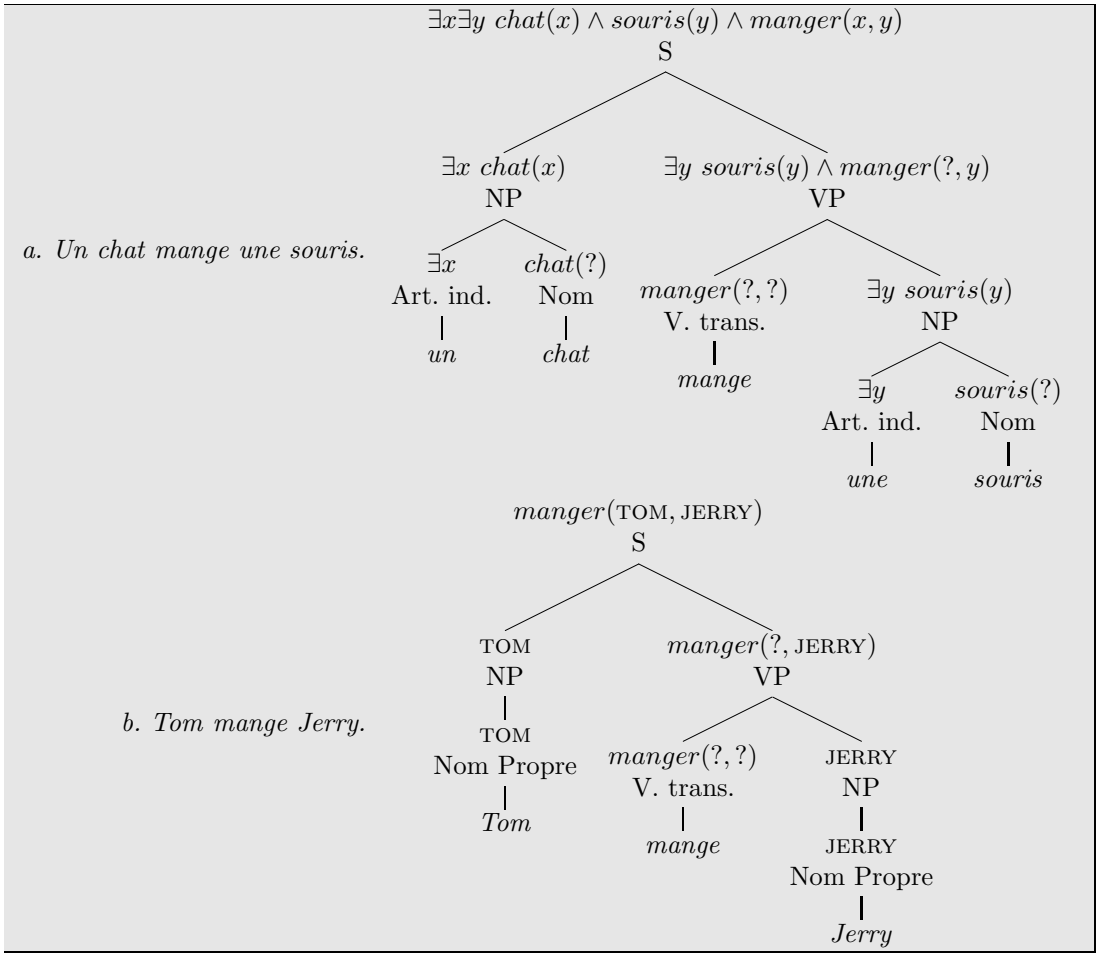


FIG. 20 – Composition de la représentation sémantique parallèlement à l'analyse syntaxique.

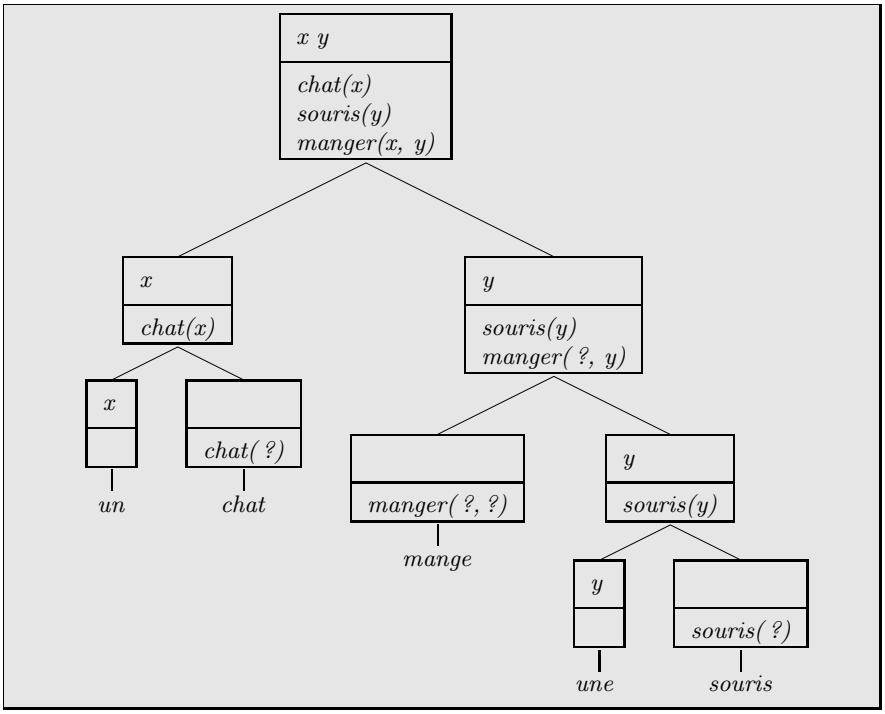


FIG. 21 – Construction d'une structure de représentation du discours (DRS).

De plus, une même règle syntaxique ne doit donner lieu qu'à un seul type de comportement sémantique, sous peine de devoir multipler les règles redondantes [15, chap. 2]. On voit par exemple que les deux phrases *"Un chat mange une souris"* et *"Tom mange Jerry"* sont sémantiquement assez proches[1]. Pourtant les représentations finales et la façon d'y parvenir présentent des différences notables. Dans le cas des noms propres (20.b), il suffit d'insérer la représentation de *"Jerry"* en seconde position du prédicat *manger* pour obtenir le syntagme verbal (VP). Reprendre cette méthode en 20.a conduirait à la formule regrettable $manger(?, \exists y\ souris(y))$. Il est cependant impensable d'associer des actions sémantiques différentes selon la "provenance" du constituant à chaque niveau syntaxique. On perdrait tout l'intérêt du regroupement en constituants substituables entre eux.

Enfin, les représentations intermédiaires (VP, NP…) sont *incomplètes*, c'est-à-dire que certaines variables sont encore à instancier (point d'interrogation dans les figures). Un autre enjeu est donc de parvenir à marquer et gérer l'information manquante.

L'introduction du lambda-calcul (λ-calcul) fournit un formalisme capable de répondre à ces problèmes. Ce formalisme est élégant et puissant, ses bases théoriques sont fortes et séduisantes pour les chercheurs intéressés par la sémantique formelle. Mais pour la conception de systèmes opérationnels dont la finalité n'est pas la représentation formelle des énoncés, sa mise en œuvre est complexe et peu robuste face à des entrées grammaticalement imparfaites. L'annexe D fournit quelques détails concernant cette approche. Les paragraphes suivants abordent d'autres types de techniques d'interprétation sémantique orientées vers des domaines et/ou des applications particulières.

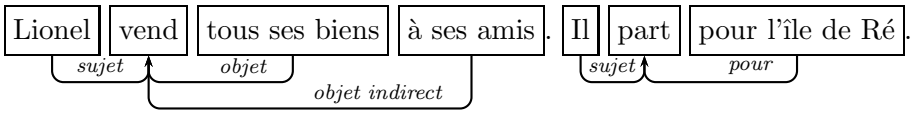
4.2.2 L'interprétation sémantique des relations grammaticales

relations grammaticales

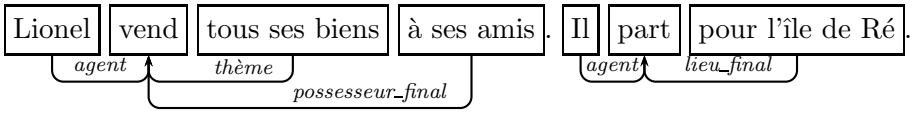
rôle thématique

Les relations grammaticales[2] s'opposent aux relations sémantiques par le fait qu'elles n'attribuent pas de rôles thématiques aux divers éléments, mais se contentent de fournir les liens syntaxiques. Les exemples suivants illustrent les relations grammaticales (29.a) et les rôles thématiques (29.b) liant les constituants d'un même discours.

(29) a.



b.



L'obtention des rôles thématiques dans le cas général nécessite des connaissances lexicales évoluées (ici, la vente est une action qui conduit à un changement de possesseur d'un objet et à un changement inverse de possesseur d'argent ; le verbe *"partir"* implique un passage d'un lieu à un autre). L'idée est donc de mettre en place un certain nombre de règles spécifiques à l'application voulue, transformant certains rôles syntaxiques précis en rôles sémantiques, et permettant d'obtenir les relations souhaitées tout en laissant de côté les constructions sans intérêt.

Le principe de conversion est relativement simple. Ainsi, les quelques règles de la figure 22 permettent de traduire la représentation a en introduisant les rôles de

[1] Même si quelques connaissances pragmatiques supplémentaires permettent de conclure que la seconde est impossible !

[2] Voir la section 3.1.3, page 11.

sujet (verbe_vente)	→	**agent**
objet (verbe_vente)	→	**thème**
objet_indirect (verbe_vente)	→	**possesseur_final**
sujet (verbe_mouvement)	→	**agent**
pour (verbe_mouvement)	→	**lieu_final**

FIG. 22 – Quelques règles d'interprétation des relations grammaticales.

REQUÊTE_VOL	→	VOL
REQUÊTE_PRIX	→	"Combien coûte" VOL " ?"
REQUÊTE_HEURE_DÉPART	→	"A quelle heure" VERBE_DÉPART VOL " ?"
VOL	→	("le" | "un") "vol" PROVENANCE DESTINATION
VOL	→	VILLE ("-" | "/") VILLE
DESTINATION	→	("à" | "vers" | "pour") VILLE
VERBE_DÉPART	→	("partir" | "décoller")
PROVENANCE	→	("à partir")? "de" VILLE
VILLE	→	"Saint-Etienne" | "Toulouse" | "Nice" | "Brisbane" | "Bujumbura" | "Tokyo" ...

FIG. 23 – Exemple de grammaire sémantique.

l'exemple b. Mais les règles de correspondance peuvent être bien plus complexes si nécessaire.

Utiliser le résultat d'une analyse syntaxique permet de s'affranchir des aléas de la construction des phrases et d'utiliser des structures déjà formalisées et moins ambiguës ; on peut alors convertir les relations grammaticales en relations sémantiques, de la façon souhaitée, avec le niveau de finesse souhaité.

4.2.3 Les grammaires sémantiques

grammaires sémantiques

Lorsque les besoins sémantiques de l'application que l'on met en œuvre sont limités et clairement définis, les *grammaires sémantiques* de *Burton* permettent de définir des règles très spécifiques, en mettant de côté des constructions moins utiles [16].

Les délimitations utilisées par l'analyseur ne sont plus celles des grammaires classiques (*syntaxiques*), comme les catégories grammaticales ou les syntagmes divers, mais des classes conçues autour d'un domaine particulier et d'une tâche précise.

Ainsi l'exemple classique de la base de données d'une compagnie aérienne est typique des besoins que peut couvrir une grammaire sémantique. La figure 23 propose une grammaire simple permettant d'analyser (entre autres) les requêtes suivantes :

(30) a. Combien coûte un vol de Saint-Etienne à Nice ? (→ REQUÊTE_PRIX)

b. A quelle heure décolle le Tokyo-Toulouse ? (→ REQUÊTE_HEURE_DÉPART)

c. Saint-Etienne/Toulouse (→ REQUÊTE_VOL)

Les mots *"Saint-Etienne"*, *"Toulouse"*, etc. ne sont pas des noms propres mais de noms de VILLE, la construction *"pour Saint-Etienne"* n'est plus un syntagme prépositionnel, mais une DESTINATION.

De cette façon, ces grammaires combinent les aspects syntaxiques et sémantiques, ainsi que les particularités de l'application, dans un même cadre. Elles permettent de

se focaliser sur les informations nécessaires, et de produire une représentation *ad hoc* de l'énoncé selon le but recherché par la suite, ce qui rend généralement inutile tout autre traitement linguistique.

4.2.4 Les patrons sémantiques

template matching

La technique des *patrons sémantiques* (ou correspondance de régions, ou surtout *template matching*) utilise généralement une analyse syntaxique de surface* pour reconnaître les groupes nominaux et verbaux de base (sans compléments), avant de détecter de simples séquences de termes prédéfinies, permettant de remplir les champs de cadres eux aussi prédéfinis. Cette approche est particulièrement appropriée aux applications d'extraction d'informations concernant un domaine délimité à l'avance. La figure 24 donne un exemple complet d'interprétation sémantique par ce moyen, avec la phrase suivante :

(31) La Finlande a battu le Canada 2 à 0 lors d'un match du premier tour du tournoi masculin de hockey sur glace.

Il est important de remarquer que cette approche de permet pas de traiter des constructions non répertoriées, mais qu'elle est robuste malgré cela. Ainsi on peut imaginer qu'un service de messagerie en ligne, proposant de la publicité ciblée, pourra analyser les messages de Lionel indiquant qu'il part pour l'île de Ré (voir l'exemple 29) et, à l'aide de quelques patrons suggérés par son client (agence de voyage), détecter l'information intéressante au milieu d'un message traitant de divers autres sujets, et proposer à l'intéressé des billets pour l'île de Ré.

4.3 Les ambiguïtés sémantiques

Si le lait cru ne convient pas à votre bébé,
faites-le bouillir.
Otto Jespersen [55].[1]

Il y a une pile de matières inflammables près de votre
voiture. Il faudra vous en débarasser.
Georges Burns And Gracie Allen, série Américaine.[2]

Les ambiguïtés sémantiques au niveau grammatical[3] sont en général des problèmes d'attribution de relations implicites (qui ne sont indiquées par des mots particuliers) entre constituants.

Chacun de ces problèmes est l'objet de nombreuses recherches, s'étendant souvent sur le plan pragmatique. Nous citons ici quelques-uns des plus fréquents.

4.3.1 L'anaphore

L'anaphore consiste à reprendre un segment du discours antérieur par un autre élément grammatical, pour éviter la répétition. Ces références sont très difficiles à repérer (à *résoudre*) automatiquement. Les cas les plus fréquents sont les *anaphores pronominales* dans lesquelles un pronom est utilisé pour remplacer un groupe nominal. Leur

[1] "If the baby does not thrive on raw milk, boil it.", cité par Jerry Hobbs [51].
[2] "There's a pile a inflammable trash next to your car. You'll have to get rid of it.", cité par Jerry Hobbs [51].
[3] Les ambiguïtés au niveau lexical sont traités à la section 4.4.

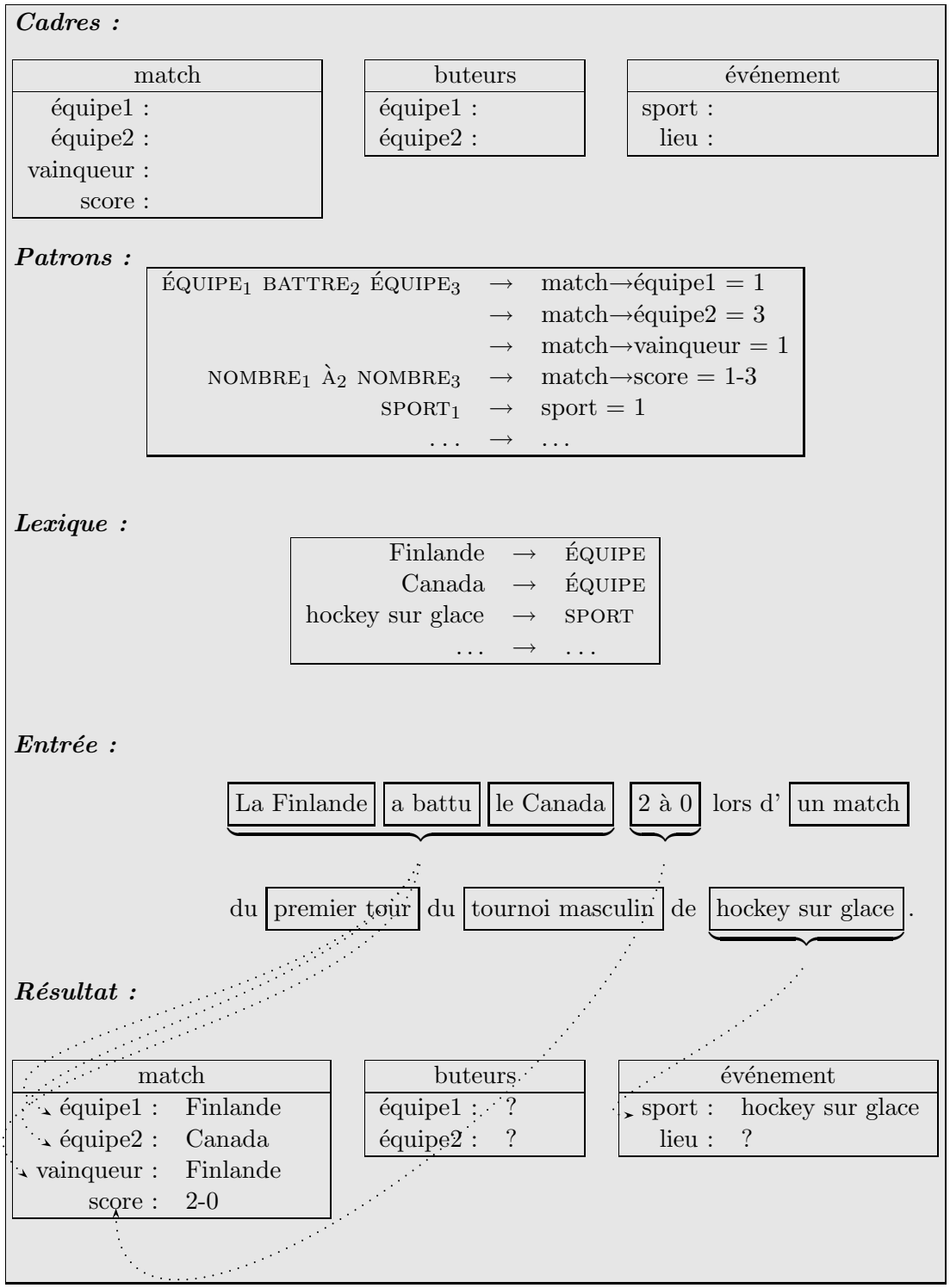


FIG. 24 – Patrons sémantiques (*template matching*).

résolution peut être simple quand il n'existe pas d'ambiguïté (exemple 32), mais se complique très vite (33, 34) jusqu'à entrer dans le domaine de la pragmatique (35).

anaphore

(32) Jacques$_1$ était furieux. Il$_1$ s'était disputé avec Georges.

(33) Dominique$_1$ rencontra Collin$_2$ à un congrès. Ils$_{1+2}$ se réconcilièrent.

(34) Nicolas$_1$ rencontra Dominique$_2$ dans un couloir. Il$_{?}$ lui$_{?}$ en voulait toujours.

(35) a. Pierre$_1$ empoisonna Sam$_2$. Il$_2$ mourut.

b. Pierre$_1$ empoisonna Sam$_2$. Il$_1$ fut arrêté.

Outre les anaphores pronominales, de nombreuses combinaisons de références sont possibles, comme l'utilisation d'adjectifs possessifs (exemple 36) ou de constituants entiers (37), mais aussi la référence à d'autres éléments que des groupes nominaux (38).

(36) La cage$_1$ du gorille$_2$ s'ouvrit. Sa$_1$ serrure devait être mal fermée.

(37) Le gorille$_1$ accélèra le pas vers le juge$_2$. Le quadrumane$_1$ avait une idée derrière la tête.

(38) Je ne peux donner la suite de l'histoire$_1$. Cela$_1$ serait pourtant délectable.

résolution des pronoms

Les procédés de résolution automatique des pronoms recourent souvent à de la connaissance du monde [90]. Les deux familles de méthodes purement sémantiques (sans connaissance extra-linguistique) les plus populaires en ce qui concerne les références à des groupes nominaux (cas le plus simple) sont l'algorithme naïf proposée par *Hobbs* [51] et la méthode des focus de *Sidner* [90][1]. Le premier suggère que le meilleur constituant candidat lors de l'occurrence d'un pronom est le dernier NP de la phrase courante ou, à défaut, le premier NP des phrases précédentes[2]. La seconde distingue le focus *acteur*, l'élément jouant un rôle actif à un moment donné du discours, et le focus du *discours*, l'élément le plus important à un moment donné du discours. Le premier est généralement le dernier constituant ayant eu le rôle d'*agent* dans la phrase, tandis que le second est choisi parmi les éléments non agents. Un pronom en position d'agent se réfèrera au focus acteur, un pronom dans une autre position sera lié au focus du discours.

focus

Ces deux approches sont illustrées à la figure 25 avec trois phrases pour lesquelles elles donnent des résultats différents, notamment l'exemple proposé par *Hobbs* [51] :

(39) The castle in Camelot remained the residence of the king until 536 when he moved it to London.[3]

4.3.2 L'ellipse

Un autre moyen de faire référence à un élément introduit au préalable en évitant la répétition est de pratiquer l'ellipse. Il s'agit d'omettre de cet élément dans une construction qui aurait dû le contenir. On peut appliquer l'ellipse à toutes sortes de constituants :

(40) a. Les Stéphanois portent des écharpes vertes et les Toulousains préfèrent les rouges et noires.

b. Les Stéphanois aiment le football et les Toulousains le rugby.

c. Les Stéphanois détestent les Parisiens et les Toulousains aussi.

[1] D'autres théories permettent de réduire les candidats possibles par des restrictions sémantiques simples, sans conclure.

[2] L'auteur dit parvenir à un taux de réussite de 88.3% en l'absence totale de connaissances ajoutées. Il rappelle néanmoins que la moitié des cas environ ne présentent aucune ambiguïté, ce qui relativise l'efficacité de la méthode.

[3] "Le chateau de Camelot demeura la résidence du roi jusqu'à 536, lorsqu'il la déplaça à Londres." Nous avons conservé la version anglaise car la traduction n'offre aucune ambiguïté grâce à la flexion féminime du pronom *"la"*, qu'on ne retrouve pas dans l'Anglais *"it"*.

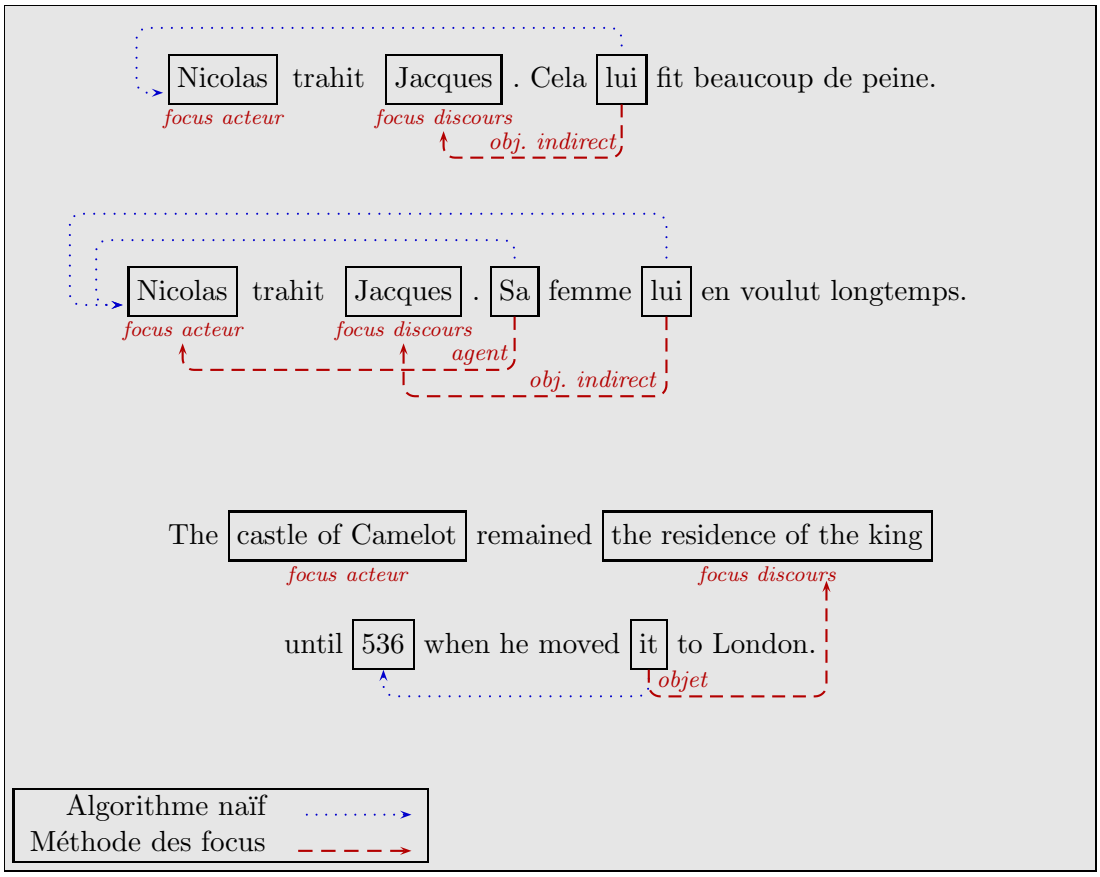


FIG. 25 – Résolution des pronoms, algorithme naïf et méthode des focus.

Remarquons qu'en dehors de tout aspect sémantique, l'ellipse pose un problème aux niveaux morphosyntaxique et syntaxique. En effet, dans l'exemple a, les adjectifs *"rouges"* et *"noires"* se comportent comme des noms, alors qu'aucune entrée lexicale ne leur attribue ce rôle. De plus, l'exemple b montre une suite de deux NP (*"les Toulousains"* et *"le rugby"*) qui constituent une proposition à part entière.

Si ces exemples sont des formes absolument correctes de la langue française, les ellipses sont surtout utilisées dans le dialogue de façon plus informelle, pour revenir sur des thèmes déjà abordés :

(41) – Je préfère aller au cinéma, et toi ? – Au théâtre.

Les concepteurs de systèmes de dialogue pour l'interrogation des bases de données ont tout particulièrement travaillé sur ce type de constructions, pour permettre des dialogues du type [49] :

(42) – Quelle est la taille du [navire] Santa Inez ?
– 127 mètres.
– Du navire nucléaire le plus rapide ?

Deux approches pour résoudre les ellipses sont utilisées. La première considère que la syntaxe de la phrase contenant une ellipse correspond (phénomène de l'ellipse mis à part) à celle de la phrase précédente [5, chap. 14]. Le "trou" provoqué par l'ellipse trouve ainsi sa correspondance. L'autre approche est celle de la proximité sémantique (avec

la méthode des grammaires sémantiques vues à la section 4.2.3). Ainsi dans l'exemple précédent, les deux suites de mots *"du Santa Inez"* et *"Du navire nucléaire le plus rapide"* sont regroupées dans la même catégorie (BATEAU). On en déduira que le reste de la première question (*"Quelle est la taille du"*) doit être ajouté à la seconde [49].

4.3.3 La portée des quantificateurs

Ce problème concerne l'ambiguïté parfois provoquée par l'introduction de quantificateurs (*tous, chaque, un*) dont la portée est incertaine, parfois même pour l'humain qui lit la phrase. Ainsi, dans les exemples suivants, un seul et unique chien suit-il tous les hommes, ou chaque homme est-il accompagné d'un chien différent ? Y a-t-il un ou plusieurs problèmes d'environnement préoccupants ? Existe-t-il un seul piano soulevé par tous les hommes, ou soulèvent-t-ils chacun leur piano ?

(43) a. Un chien suit chaque homme qui passe la porte du bar.
b. Un problème d'environnement préoccupe tous les politiciens sérieux.
c. Tous les hommes soulevèrent un piano.

Si ces exemples ont l'avantage d'être faciles à comprendre car également ambigus pour l'homme, il faut noter que les phrases suivantes, de constructions symétriques à celles des premières, sont tout aussi ambiguës pour une machine [101] :

(44) a. Une femme a dit de chaque sénateur qu'il était malade.[1]
b. Ron a parlé d'un problème à chaque femme.[2]
c. Chaque mathématicien parle une langue étrangère.[3]

Encore une fois, cette problématique est particulièrement importante pour les interfaces d'interrogation de bases de données en langage naturel. Les solutions proposées se situent au niveau sémantique, nous ne les détaillerons pas ici.

4.4 La sémantique lexicale

Ici la mer est bleue et les nuits sont blanches
Publicité pour le Club Med.

Au lieu d'acheter une voiture, achetez une SAAB
Publicité pour SAAB.[4]

Les sections précédentes ont traité de la sémantique grammaticale, en considérant plus les catégories que les mots eux-mêmes. Pour prendre un seul exemple, nous avons représenté le mot *"bureau"* dans la section 4.1.3 par le prédicat *bureau(x)*, sans nous préoccuper de distinguer les différents sens que pouvait prendre ce mot (la pièce ou le meuble en particulier[5]).

De plus, connaître les propriétés des concepts ou des objets désignés par les mots peut constituer un moyen de limiter les ambiguïtés de tous types. Ainsi le rattachement du pronom *"it"* dans l'exemple 39 devient immédiat si l'on sait que *"move"* implique un déplacement et que ni un nombre (*"536"*) ni un château ne sont déplaçables. De la

[1] "Some woman said every senator was sick." [101]
[2] "Ron talked to each woman about a problem." [101]
[3] "Every mathematician speaks a foreign language." [101]
[4] Ces utilisations de la polysémie (premier exemple) et de l'hypéronymie (second exemple), ainsi que de nombreuses autres, ont été collectées et analysées par Michel Ballabriga [10].
[5] Mais le *Trésor de la Langue Française* distingue plus de 20 sens différents pour la graphie *"bureau"*.

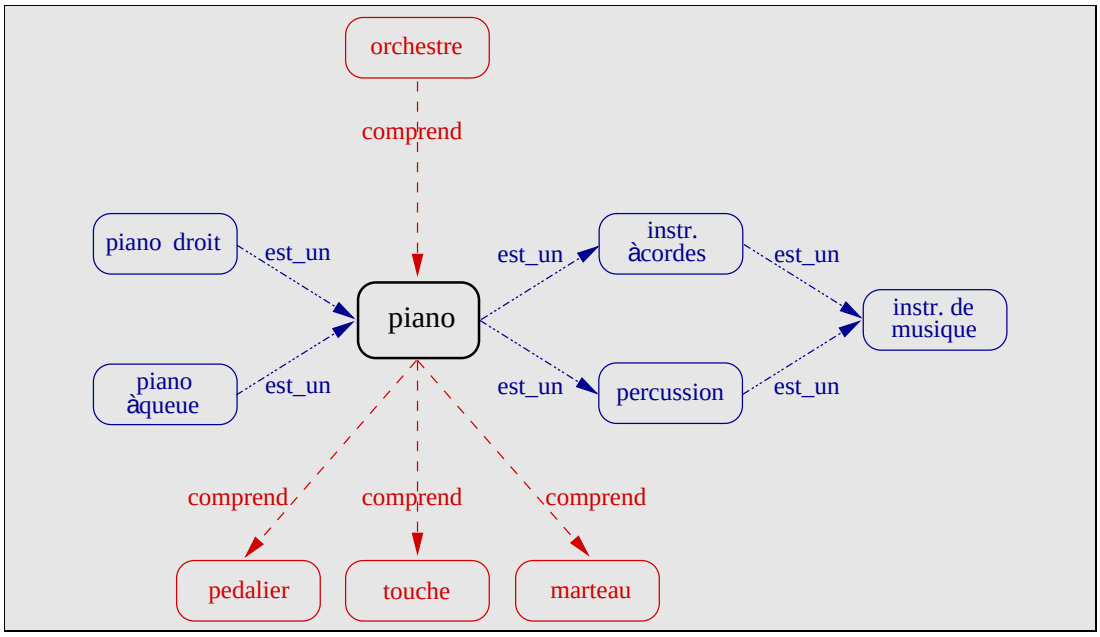


FIG. 26 – Graphe de relations lexicales centré sur le terme *"piano"*. Dans le cas général, les relations d'holonymie (axe vertical) s'héritent par la relation d'hyponomie (axe horizontal). Ainsi les pianos droits et les pianos à queue ont (entre autres) les composants indiqués pour le piano. Une autre dimension pourrait être celles des *propriétés*, qui s'héritent elles aussi (par exemple, un instrument de musique émet du son).

même façon, l'anaphore 37 est résolue facilement en sachant qu'un gorille est une sorte de quadrumane. Enfin l'adjectif possessif *"sa"* de l'énoncé 36 est immédiatement relié à la cage si l'on a la connaissance nécessaire (une cage possède une serrure).

Ces exemples illustrent quelques relations importantes pouvant exister entre les mots d'un lexique :

polysémie
- la *polysémie* et l'*homonymie* sont la propriété de certaines formes graphiques (signifiants) de renvoyer à plusieurs sens (signifiés) différents (*"bureau"* est une forme polysémique)[1].

synonymie
- la *synonymie* est le lien entre deux mots ayant le même sens.

hyponymie
- l'*hyponymie* est la relation d'inclusion entre deux mots dont l'un (l'*hyponyme*) est plus spécifique que l'autre (l'*hyperonyme*). Ainsi le gorille est un hyponyme du quadrumane, la fleur est un hypéronyme de la tulipe.

méronymie *holonymie*
- la *méronymie* et l'*holonymie* expriment la relation de partie à tout : la serrure est une partie (méronyme) de la cage. De même, le bâtiment contient (est un holonyme de) une pièce.

Il est possible de représenter les relations lexicales entre les mots par un graphe, dont une petite portion est donnée à la figure 26, centrée sur la notion de *"piano"*. Des bases répertoriant les différentes relations existant entre les mots ont été créées, la plus connue et la plus utilisée (pour le domaine général et la langue anglaise) étant

WordNet
WordNet [70].

[1] La différence entre les deux notions réside dans le fait que l'homonymie lie deux signifiés sans aucune relation sémantique (comme *"son"* – le bruit ou le terme de meunerie) tandis que les termes polysémiques comportent des traits en commun (comme le *"bureau"*).

La section 5.3 développe les problèmes spécifiques que posent ces différents phénomènes en recherche d'information.

4.5 Vers la pragmatique

On peut prouver que le progrès social est bien meilleur avec du sucre.
Eugène Ionesco, *La Cantatrice Chauve*

pragmatique

Le terme de *pragmatique* regroupe un grand nombre de domaines de recherche, englobant en fait tous les problèmes qui ne peuvent pas être résolus avec la syntaxe et la sémantique. A partir du constat qu' *"un énoncé ne peut prendre tout son sens que si on le replace dans son milieu d'origine"*, elle s'attache à *"l'étude de l'environnement d'une phrase, au moment où elle est émise"* [54, p. 69]. Elle implique l'utilisation de connaissances extra-linguistiques sur le contexte du discours (phrases précédentes, situation et histoire commune des personnages, etc.) et sur le monde en général.

Les études à ce sujet supposent un apport important de connaissances, mais également une analyse fine des phénomènes impliqués. Les applications pratiques sont rares. Cependant il nous semble utile d'énumérer quelques-uns des domaines concernés pour terminer notre tour d'horizon des enjeux de l'analyse du langage et pour souligner le chemin qui reste à parcourir.

La déictique est l'ensemble des allusions directes des interlocuteurs au contexte de l'énonciation : il peut s'agir de références à des entités déjà introduites (comme les *anaphores* déjà abordées) ou supposées connues par les interlocuteurs (comme le lieu et l'heure courante, phrase 45) ou même par exemple montrées avec le doigt dans le langage oral (*déictique gestuelle*, phrase 46).

(45) J'ai rencontré la reine d'Angleterre *ici même*. C'était *l'année dernière*.

(46) Séparons-nous. Je pars avec *toi, toi et toi*, et *vous* partez de votre côté.

Les implicatures conversationnelles concernent ce que l'on peut déduire d'une phrase énoncée en dehors de sa signification littérale. Dans les dialogues qui suivent, la phrase prononcée par B comporte des implications de ce type, dont la signification est résumée après.

(47) A : Le voisin est-il chez lui ?
B : Sa voiture est devant le portail.
⇒ le voisin est probablement chez lui, car sa voiture est devant le portail.

(48) A : Je suis en panne d'essence.
B : Il y a un garage à deux pas d'ici.
⇒ Il y a un garage près d'ici, vous pourriez y aller pour alimenter votre voiture en essence.

(49) A : Georges et Donald vont-ils s'arrêter là ?
B : Sont-ils des hommes modérés ?
⇒ Georges et Donald ne vont bien sûr pas s'arrêter là !

Les présuppositions regroupent les informations que l'on peut déduire sur le contexte lors de l'énonciation de certaines phrases. Encore une fois, contentons-nous d'exemples (ce qui suit la flèche est la présupposition) :

(50) Le Roi de France est sage.
⇒ il existe un Roi de France.

(51) Jean regrette d'avoir fait ses études à Toulouse.
⇒ Jean est une personne identifiable par le locuteur et le destinataire du message. Jean a fait ses études à Toulouse.

(52) Si le vice-chancelier invite Simone de Beauvoir, il regrettera d'avoir une féministe à sa table.[1]
⇒ Simone de Beauvoir est une féministe.

(53) Si le vice-chancelier invite le président des États-Unis, il regrettera d'avoir une féministe à sa table.[1]
⇒ le vice-chancelier a également invité une féministe.

Pour avoir plus de détails sur ces problèmes et d'autres (actes de langage, métaphores, structure de la conversation, etc.), le lecteur peut consulter les ouvrages spécialisés [63, 27].

5 Le traitement de la langue dans la recherche d'information

La contribution du traitement automatique de la langue (TAL) dans la recherche d'information peut sembler modeste au vu de la problématique générale de l'analyse du langage naturel que nous avons abordée dans les sections précédentes. Il semble en effet illusoire, dans l'état actuel des technologies, d'imaginer une véritable "compréhension" du langage par une machine, ne serait-ce qu'en termes de masse de connaissance nécessaire et de temps passé à manipuler cette connaissance [5, chap. 15].

Malgré cela, de nombreux aspects de la linguistique informatique sont utilisés en recherche d'information, avec plus ou moins de succès, que ce soit aux niveaux morphologique*, syntaxique* ou sémantique* [61]. Les systèmes utilisant des bases de connaissance importantes se sont, quant à eux, cantonnés à des applications très spécifiques dans des domaines particuliers [62].

5.1 Variations morphologiques

Les traitements morphologiques appliqués à la RI visent tous à opérer des regroupements de mots ayant la même base sémantique, mais ayant subi une flexion* ou une dérivation*. Ces techniques peuvent être utilisées pour regrouper des mots autour d'une base commune (lors de l'indexation) ou à l'inverse pour étendre une requête avec tous les mots dérivés susceptibles d'apparaître dans les documents. Dans les deux cas le but est d'augmenter le rappel* en considérant plus de mots. Le danger est bien sûr de diminuer la précision* si des mots sémantiquement éloignés sont traités par erreur.

stemming

La racinisation (ou *stemming*) la plus courante utilise une approximation des phénomènes linguistiques d'une langue donnée, comme les mécanismes habituels de conjugaison, d'accords en genre et en nombre, ou de dérivation, et tente de supprimer les suffixes tout en regroupant les différents allomorphes (variantes graphiques d'une même racine, comme *"produi-re"* et *"produc-teur"*, ou *"ir-ait"* et *"all-ons"* [98, 44]). Les algorithmes les plus courants pour ce travail sont ceux de Lovins [64] et de Porter [80] (Jacques Savoy pour le Français [85, 86]). Notons que généralement la racinisation d'un mot par ce moyen ne conduit pas à un mot existant (par exemple, la racinisation des mots *"relativisme"* et *"relativisation"* conduiront tous deux à la pseudo-racine *"relativ"*). De plus des mots aux sens éloignés peuvent se voir attribuer une racine commune par ces algorithmes, comme par exemple "cheval", "chevalerie", "chevalier" et "chevalet" [87], avec des effets indésirables en termes de précision. D'autres méthodes tentent d'étendre les requêtes avec des termes morphologiquement proches, acquis de manière plus élaborée [71].

[1] Exemple de Karttunen cité par [63]

D'autres algorithmes utilisent la catégorie grammaticale (étiquetage morphosyntaxique*), aidés de dictionnaires ou de connaissances linguistiques plus évoluées, pour effectuer une analyse flexionnelle et dérivationnelle plus fine. Ainsi, Flemm [73] utilise un analyseur morphosyntaxique (Brill Tagger [18] ou TreeTagger [89]) et un système à base de règles et d'exceptions pour découvrir quelles opérations morphologiques ont conduit au mot proposé.

L'efficacité des algorithmes de *stemming* est généralement évaluée de façon indirecte, en observant l'effet de leur utilisation sur les résultats des moteurs de recherche. Cette efficacité dépend beaucoup de la langue étudiée [9] et est dans la plupart des cas assez limitée [72]. Récemment une comparaison des algorithmes les plus utilisés en Anglais a été effectuée indépendamment du processus de recherche d'information [28].

5.2 Variations syntaxiques

Les connaissances syntaxiques utilisées en pratique par la recherche d'information sont relativement modestes, et se cantonnent généralement à l'étude des syntagmes nominaux*. On peut distinguer :

- Les termes "complexes", des lemmes* formés de deux ou plusieurs termes, qui sont plus précis et moins ambigus que les termes séparés (comme *"pomme de terre"*). Des techniques ont été mises en place pour indexer les termes complexes, par des méthodes statistiques (co-occurrences de termes dans les corpus) ou syntaxiques (ensembles de patrons prédéfinis) [72].
- Les termes "structurés", que l'on ramène aux relations de dépendance existant entre les termes qu'ils contiennent. Ainsi, les suites de mots *"abattage d'arbre"* et *"les arbres ont été abattus"* peuvent être ramenés à une forme commune de type *"arbre+abattre"* où *"arbre"* est la tête du syntagme. On peut obtenir les deux formes (et d'autres) par application de règles de transformations syntaxiques particulières [35, 95].

L'application de ces différentes techniques à la recherche d'information nécessite des adaptations profondes des méthodes d'indexation et de pondération des termes [72], pour des résultats variables, souvent positifs mais rarement décisifs.

5.3 Variations sémantiques

Les variations sémantiques considérées en recherche d'information sont essentiellement d'ordre lexical. Les différents phénomènes lexicaux décrits à la section 4.4 peuvent être regroupés en deux classes de problèmes pour la recherche d'information :

- Les relations liant des lexèmes différents mais sémantiquement proches. La synonymie* et l'hyponymie* ont en effet pour conséquence que les termes utilisés dans une requête ne sont pas les mêmes que ceux employés dans des documents en rapport avec cette requête[1], ce qui nuit à la précision*.
- La multitude de sens que peut prendre une forme graphique donnée (polysémie*, homonymie). Le résultat en recherche d'information est qu'un terme employé dans une requête peut se retrouver avec des sens éloignés dans des documents non pertinents[2,3]. Dans ce cas c'est le rappel* qui est touché.

[1] Ainsi *Furnas et al.* [42] estime entre 7 et 18 % (selon le domaine, en Anglais) la probabilité pour que deux personnes attribuent le même mot à un objet donné.

[2] Dans la même étude [42], la probabilité pour que deux personnes utilisant un terme donné désignent le même objet est estimée entre 15 et 73 %.

[3] Le Petit Robert annonce ainsi, dans son édition de 2006, un total de 60000 mots et de 300000 sens, soit en moyenne 5 sens par mot, et bien plus pour les mots courants.

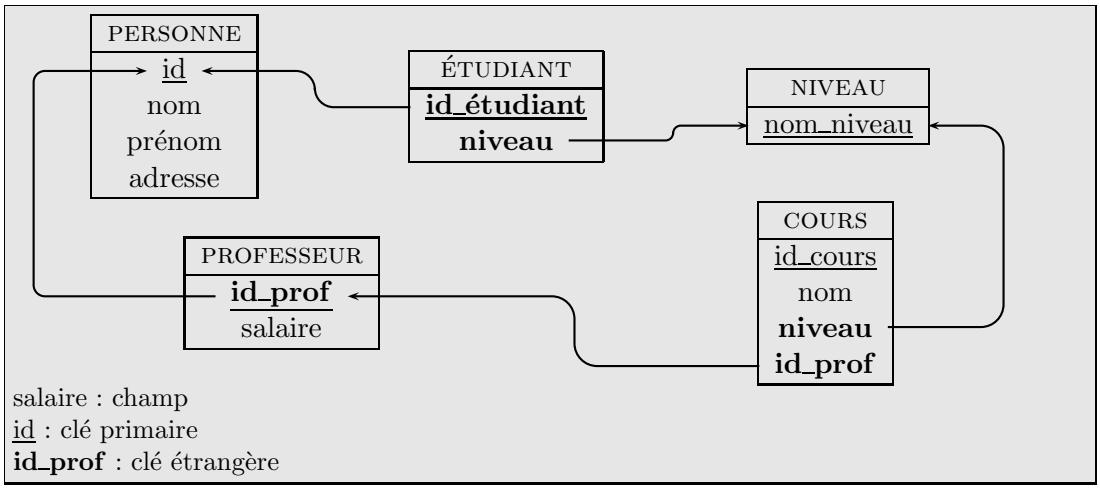


FIG. 27 – Exemple de tables de bases de données.

L'utilisation du premier phénomène permet soit d'enrichir les requêtes en utilisant les mots ayant des sens approchés ou en liaison avec les mots initiaux, soit d'effectuer une indexation* plus complexe basée sur les concepts plutôt que sur les mots eux-mêmes. La base WordNet [70] permet ce travail pour des corpus généralistes. Il est aussi possible d'acquérir automatiquement des lexiques sémantiques à partir de corpus [23]. L'utilisation de telles bases de façon automatique se heurte aux difficultés de sélection des relations appropriées et à une accentuation du problème de la polysémie. Il est alors indispensable de mettre en place des mécanismes pour "désambiguïser" les termes polysémiques [9]. La plupart des études sur le sujet [84, 102, 103, 46, 93] aboutissent à des résultats peu convaincants ; notamment les résultats des expérimentations utilisant la base WordNet [36] sont assez mitigés [104].

6 Les interfaces de requêtes en langage naturel pour les bases de données

Une autre application du TAL qui nous intéresse particulièrement est la conception d'interfaces en langage naturel pour l'interrogation des bases de données.

6.1 Intérêts

L'accès aux bases de données est en général effectué à l'aide d'un langage structuré, le plus souvent SQL [37]. La syntaxe de ce langage nous importe peu ; un exemple en est toutefois donné à la figure 28, pour donner une idée de sa complexité. La requête concerne les tables décrites à la figure 27.

Les avantages d'un système permettant à l'utilisateur de formuler sa requête en langage naturel plutôt qu'en SQL sont évidents. *Hendrix* [48] donne un bon aperçu des capacités attendues d'un tel système, des aspects techniques (accès multiples) jusqu'aux finesses d'analyse (comme la correction orthographique). Une interface en langage naturel évite surtout l'apprentissage d'un langage formel complexe, ainsi que la connaissance rigoureuse de la structure de la base [81]. Elle permet également le *dialogue* avec l'humain [20], et ainsi, notamment, la référence à des questions déjà posées.

Qui enseigne les mathématiques à Dupont ?

```
SELECT P2.nom, P2.prénom
FROM PERSONNE P1, PERSONNE P2, ETUDIANT E, COURS C
WHERE C.niveau = E.niveau
  AND E.id_étudiant = P1.id
  AND P1.nom = "Dupont"
  AND C.nom = "Mathématiques"
  AND C.id_prof = P2.id
```

FIG. 28 – Exemple de requête en SQL pour une base de données.

Un exemple d'un tel dialogue pourrait être :

1> *Qui enseigne les mathématiques ?*
→ `Taberly, Meallares, Robert, Filustier`
2> *Lequel enseigne en $2^{ème}$ année ?*
→ `Robert`
3> *En $1^{ère}$ année ?*
→ `Filustier`
4> *Tous les enseignants ont-ils un cours ?*
→ `oui.`

Ces échanges donnent un aperçu des possibilités offertes pour améliorer le confort d'utilisation d'un système de gestion de base de données ; les trois dernières questions illustrent également des problèmes qui se posent de façon particulièrement aiguë dans ce domaine, et qui ont été décrits précédemment[1] : les références anaphoriques* (question 2), les ellipses* (question 3) et la portée des quantificateurs (question 4 [2]).

Les autres problèmes "classiques" d'ambiguïté, également abordés plus haut dans ce chapitre[3], ne sont pas de reste. Il est important de noter que toute erreur d'analyse, à quelque niveau que ce soit, conduit à une interprétation erronée de la requête, et donc à un résultat totalement faux.

6.2 Architectures

Les choix importants concernant l'architecture d'une interface pour les bases de données sont d'une part la place accordée à la syntaxe des énoncés, et d'autre part le poids et le mode de représentation de la sémantique.

Ces choix influent sur les caractéristiques du système, notamment en termes de couverture du langage, de portabilité vers une autre base, de robustesse.

De nombreuses architectures ont été mises en œuvre, nous les avons séparées en trois grands groupes : les patrons sémantiques, les grammaires sémantiques et l'utilisation de langages de représentation intermédiaires.

[1] Section 4.3, page 27.
[2] Une machine peut en effet interpréter que l'on demande s'il existe un cours qui est donné par tous les enseignants, ou bien (plus probablement) si chaque enseignant s'est vu attribué au moins un cours.
[3] Section 3.2, page 11.

Patrons :

"enseigne" ... MATIÈRE	→	`SELECT P.nom FROM COURS C, PERSONNE P WHERE C.nom =` MATIÈRE `AND C.id_prof = P.id`
"matières" ... NIVEAU	→	`SELECT C.nom FROM COURS C WHERE C.niveau =` NIVEAU
...	→	...

Lexique :

`SELECT DISTINCT nom FROM COURS`	→	MATIÈRE
`SELECT DISTINCT nom_niveau FROM NIVEAU`	→	NIVEAU

FIG. 29 – Patrons sémantiques pour l'interrogation de bases de données.

6.2.1 Les patrons sémantiques

patrons sémantiques

Nous avons montré précédemment[1] un exemple d'utilisation des patrons sémantiques pour l'extraction d'information. Il est aussi possible de les appliquer aux requêtes en langage naturel [7]. La syntaxe est alors presque absente du système.

Par exemple, les patrons simplistes de la figure 29 permettent d'indiquer les actions à mettre en œuvre si le mot *"enseigne"* apparaît suivi par un nom de matière (→ sélection des enseignants pour cette matière), ou le mot *"matières"* suivi par un niveau (→ sélection des matières pour ce niveau). Remarquons que le lexique n'est plus (seulement) composé d'une liste de mots, mais dépend également du contenu de la base de données (ainsi les noms de matières sont enregistrés dans la table COURS).

Ainsi les deux requêtes simples qui suivent sont analysées par les deux patrons de la figure 29 :

(54) Qui enseigne les mathématiques ?

(55) Quelles matières sont enseignées en 1ère année ?

L'avantage essentiel de cette approche est sa simplicité (même si bien entendu les patrons sont en réalité plus complexes que ceux indiqués ici). Cependant de nombreuses constructions linguistiques ne peuvent être traitées, et surtout des énoncés peuvent être mal interprétés.

6.2.2 Les grammaires sémantiques

grammaires sémantiques

Nous avons décrit les grammaires sémantiques à la section 4.2.3 (page 26), en donnant un exemple s'adaptant bien aux interfaces de bases de données. L'avantage de ces grammaires est que la représentation arborescente qu'elles produisent est bien adaptée au domaine traité, et donc que le lien avec les objets de la base de données et la construction de la requête formelle sont facilités.

Un nouvel exemple permettant de traiter notamment les exemples 54 et 55 de la section précédente est traité aux figures 30 et 31 (arbre sémantique, que nous pouvons comparer avec l'arbre syntaxique de la même phrase, présenté plus loin – figure 32 – pour constater qu'ils ne correspondent ni par les catégories, ni par la structure).

[1] Section 4.2.4, page 27.

REQUÊTE_ENSEIGNANT	→	PERSONNE_ENSEIGNANT ENSEIGNE_ACTIF MATIÈRE
REQUÊTE_MATIÈRE	→	”Quelles matières” ENSEIGNE_PASSIF ”par” ENSEIGNANT
REQUÊTE_MATIÈRE	→	”Quelles matières” ENSEIGNE_PASSIF ”en” NIVEAU
PERSONNE_ENSEIGNANT	→	(”quel enseignant” | PERSONNE_QUESTION)
PERSONNE_QUESTION	→	”qui” | ”quelle personne”
ENSEIGNE_ACTIF	→	”enseigne” | ”donne le cours de” ...
ENSEIGNE_PASSIF	→	”être enseigné” ...
ENSEIGNANT	→	(`SELECT DISTINCT id_prof FROM PROFESSEUR`)
COURS	→	(`SELECT DISTINCT nom_niveau FROM NIVEAU`)
MATIÈRE	→	(”la” | ”le” | ”les”)? (`SELECT DISTINCT nom FROM COURS`)

FIG. 30 – Grammaire sémantique pour l'interrogation de bases de données.

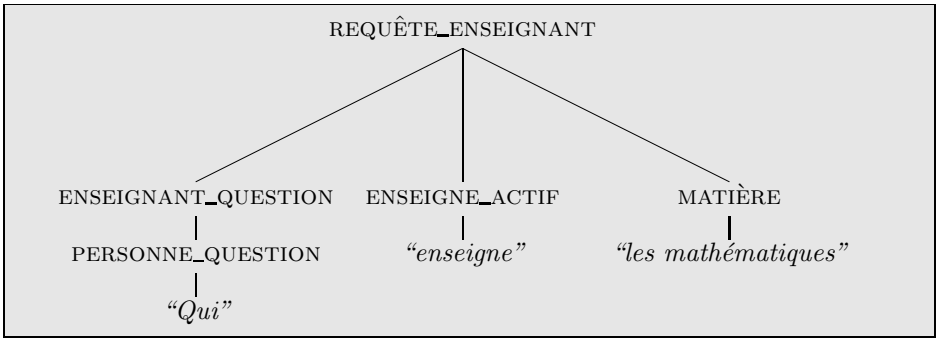


FIG. 31 – Représentation arborescente de l'analyse par une grammaire sémantique.

Une explication très détaillée du fonctionnement d'un tel système est la description de LADDER par *Hendrix et al.* [49]. Les limites y sont également clairement exposées. Parmi celles-ci, on peut citer :

- le manque de robustesse face à des constructions non prévues (notamment des assertions[1]) ;
- l'irrégularité de la couverture (les règles étant très spécifiques, un phénomène peut être très bien couvert tandis qu'un autre est délaissé[2]) ;
- les ambiguïtés syntaxiques et sémantiques ;
- la difficulté de porter l'application dans un nouveau domaine. Pourtant, le système LADDER tente de limiter ce problème en modulant les différentes étapes de l'analyse ; mais la non-portabilité est une contrainte indissociable de la solution des grammaires sémantiques. En effet, les informations lexicales, syntaxiques et structurelles sont combinées dans le même ensemble de règles, sans distinction.

[1] Les assertions sont des phrases déclaratives, les plus courantes dans le langage de tous les jours, mais peu employées pour l'interrogation des bases de données, où la forme interrogative est plus appropriée.

[2] Les auteurs donnent l'exemple de la voix passive, qu'il est nécessaire de modéliser pour chaque construction impliquant des verbes. Ce n'est pas le cas des systèmes basés sur les relations grammaticales comme Lunar, qui se place à un niveau d'abstraction supérieur.

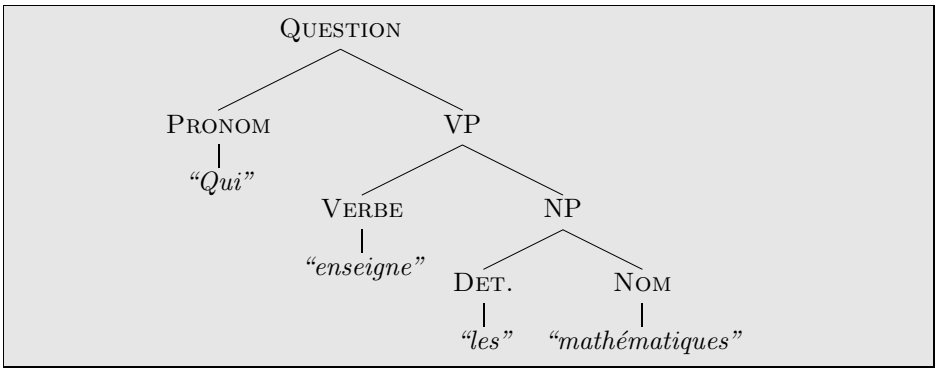


FIG. 32 – Représentation arborescente obtenue par l'analyse syntaxique.

6.2.3 Les langages de représentation intermédiaires

L'utilisation d'un ou plusieurs langages intermédiaires de représentation de la requête est un choix très commun. Elle permet de séparer l'analyse en deux parties, dont la première est indépendante de l'application.

Le principe, inspiré du système Lunar [109], l'un des pionniers en la matière, est d'effectuer une analyse syntaxique "classique" de la requête, à l'aide d'une grammaire générale, pour obtenir une première représentation formelle de l'énoncé (figure 32). La méthode employée pour traiter cette forme intermédiaire varie alors selon les systèmes, mais le principe général est d'incorporer séparément les connaissances concernant la structure et/ou le domaine de l'application.

Dans Lunar, IRUS [12] ou MASQUE [6], chaque mot de la requête est transformé en un morceau de relation logique, le tout permettant d'aboutir à une expression logique applicable à la base de données[1]. Par exemple, dans Lunar, le pronom "which" introduit l'expression `FOR EVERY X ... PRINTOUT X`, indiquant que tous les éléments d'un type donné doivent être retournés à l'utilisateur. Les questions fermées introduisent le mot-clé `TEST`. En joignant les expressions, on obtient la requête finale. Ainsi, la question *"Does S13004 contains Europium in plag ?"* est traduite en :

```
(TEST (CONTAIN (NPR = X3 / (QUOTE S13004)) (NPR = X4 / QUOTE EU))
                    (NPR = X5 / QUOTE PLAG))))
```

D'autres systèmes proposent une analyse sémantique profonde, liée à l'analyse syntaxique (analyse compositionnelle[2]). C'est le cas de JANUS [50] ou de Chat-80 [106].

Les connaissances qu'il est nécessaire d'ajouter dans ce type d'approches ont une forme très différente de celles des patrons sémantiques par exemple. Dans ce dernier cas, on énumère les constructions possibles concernant l'application, et on propose une interprétation en termes d'interrogation de la base. Ici, l'énoncé lui-même étant traité à part, les connaissances sont insérées de façon beaucoup plus conceptuelle, par un thésaurus, par exemple[3], ou une modélisation du domaine plus approfondie, comme illustré à la figure 33 pour une base de données concernant les cours d'une université [25].

[1] Ce qui s'approche de la technique d'interprétation des relations grammaticales décrite à la section 4.2.2, page 25.

[2] Voir la section 4.2 page 23.

[3] Comme présenté à la section 4.4, page 31.

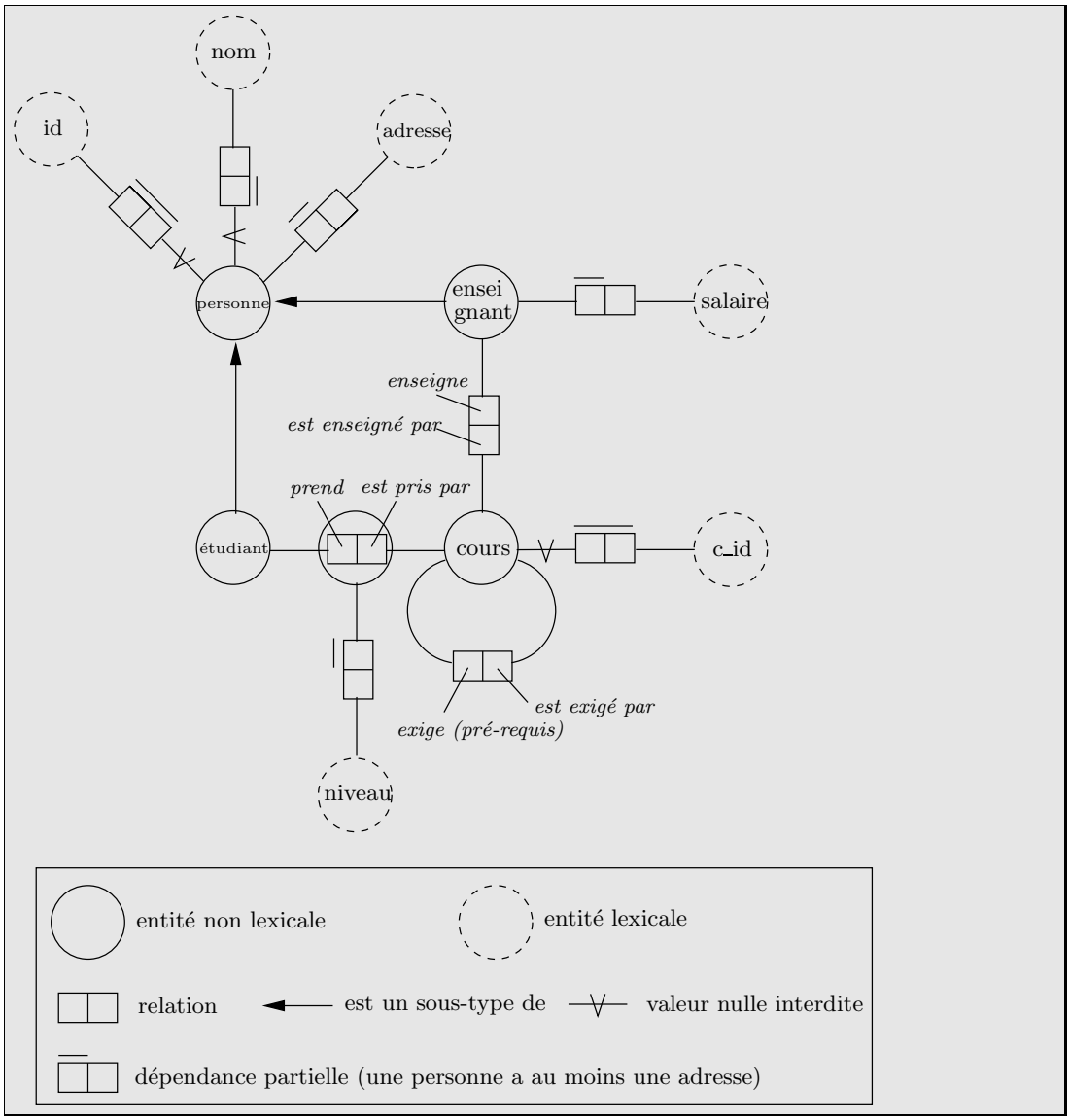


FIG. 33 – Modélisation du domaine de la base de données (selon *Copestake et Sparck Jones [25]*).

Les systèmes utilisant des représentations intermédiaires ont l'avantage de simplifier la portabilité de l'interface, puisqu'une grande partie du processus est indépendante du domaine d'application. Ceci ne signifie pas qu'un changement du type de données est facile ; il est cependant plus localisé. Les deux inconvénients majeurs sont d'une part que l'obtention d'une représentation intermédiaire ne garantit pas le succès de la traduction finale (en quelque sorte, chaque intermédiaire est un risque de plus de se tromper), et d'autre part qu'il est difficile de transformer cette représentation en un langage non conçu dans ce but, comme le standard SQL. C'est pourquoi nombre de systèmes utilisent leur propre langage formel d'interrogation, ce qui nécessite de mettre en place un système de gestion de la base de données propre à l'application.

6.3 Inconvénients et limites des interfaces

Le premier problème pointé par les utilisateurs des interfaces est que le langage n'est pas si naturel qu'on pourrait s'y attendre, et que l'usage du système nécessite un apprentissage des structures acceptées ou refusées par l'interface. La couverture n'est pas toujours évidente à comprendre, c'est-à-dire que les raisons pour lesquelles une requête est bien analysée et une autre ne l'est pas sont parfois obscures [7]. Une étude de *Dekleva* [29] montre que, sans entraînement et sans aucune modification, 53,8 % des questions sont traitées correctement (par le système INTELLECT, une interface commerciale parmi les plus connues).

De plus, lorsque l'interface échoue dans l'analyse d'une requête, il est difficile de savoir si l'erreur vient d'un problème de couverture *linguistique* (qui peut être corrigée en reformulant la demande) ou de couverture *conceptuelle*, c'est-à-dire une question concernant des informations non contenues par la base.

Une autre cause d'égarement de l'utilisateur est que celui-ci s'imagine parfois avoir affaire à un système *intelligent*, capable de raisonnement. Il pose alors des questions faisant appel à du sens commun ou à des capacités de déduction que la machine ne possède absolument pas.

Enfin, il arrive que le système renvoie une réponse erronée, due à une mauvaise interprétation de la requête. Dans ce cas, le risque est que l'utilisateur ne se rende pas compte que la sortie du système ne correspond pas à son besoin.

Ces problèmes de relation avec la clientèle sont accompagnés d'un obstacle technique. En effet, la plupart des systèmes d'interrogation en langage naturel ne sont pas portables, c'est-à-dire qu'il est impossible, ou très difficile, d'appliquer une interface à un nouveau domaine de connaissance ou à un nouveau système de gestion [25].

Des recherches ont tenté de réduire certains de ces défauts, avec une reformulation de la requête, un diagnostic d'erreur ou des modules génériques, mais ces écueils ont empêché le développement commercial à grande échelle des interfaces en langage naturel. La conclusion de *Capindale et Crawford* [19] (cités par *Androutsopoulos et al.* [7]) est que *"le langage naturel est une méthode d'interaction efficace pour les utilisateurs novices ayant une bonne connaissance de la base de données, qui demandent des réponses précises dans un domaine limité"*[1].

7 Les interfaces en langage naturel pour la recherche d'information semi-structurée

Lancées en 2003 avec la première campagne d'INEX sur le sujet, les recherches concernant les interfaces de requêtes en langage naturel (ILN) pour la recherche d'information semi-structurée n'en sont qu'à leurs premiers pas.

La RI structurée combinant des aspects des bases de données avec ceux de la recherche d'information, il va de soi que les ILN pour ce domaine mêlent également les intérêts et les particularités de ces deux spécialités. S'il n'existe pas vraiment d'interfaces de ce type pour la recherche d'information traditionnelle (il s'agit plutôt d'extraction de mots-clés – voir la section 5), nous avons vu à la section précédente que les recherches avaient été très actives à ce niveau pour l'accès aux bases de données.

Cette dernière section introduit ce nouvel axe de travail, en mettant en avant ses motivations et ses spécificités et en décrivant de façon plus détaillée la tâche "Langage

[1] "Natural language is an effective method of interaction for casual users with a good knowledge of the database, who perform question-answering tasks, in a restricted domain."

Naturel" d'INEX. Nous examinons également les deux seules approches (outre la nôtre) qui ont été l'objet de communications.

7.1 Motivation

Certaines motivations présentées ici sont également importantes dans le cadre des bases de données. Cependant nous allons voir que les ILN pour les collections XML correspondent à des besoins différents et possèdent des caractéristiques différentes.

7.1.1 Complexité des langages de requêtes

La première motivation qui pousse à s'intéresser à la conception d'interfaces en langage naturel vient du fait que l'expression du besoin d'information dans un langage structuré, possédant une syntaxe et une sémantique formalisées, est trop complexe pour la majorité des utilisateurs. *O'Keefe et Trotman* [76] ont étudié ce problème avec cinq langages de requêtes structurés (HyTime, DSSSL, CSS, XPath, XIRQL) et ont conclu qu'aucun d'eux n'apportait la simplicité d'utilisation nécessaire.

A titre d'exemple, nous pouvons citer le cas de la campagne INEX. Pour cette campagne, les requêtes (ou *topics*) sont écrites par les participants eux-mêmes, c'est-à-dire des experts en RI, habitués à manipuler des langages de requêtes [97]. Comme nous l'avons déjà signalé, en 2003, le langage XPath [105] était utilisé. 63 % des requêtes proposées en premier lieu comportaient des erreurs syntaxiques ou sémantiques majeures. Pas moins de 12 tours de corrections ont été nécessaires pour aboutir à un ensemble de topics satisfaisants.

Cette situation a conduit à l'adoption en 2004 de NEXI, un langage très simplifié et beaucoup moins expressif. Le taux d'erreurs initial est tombé à 12 % et le nombre de révisions nécessaires a chuté. Pourtant ces chiffres restent très élevés pour des experts.

De plus, *van Zwol et al.* [100] a confirmé en 2005 par une étude auprès d'utilisateurs novices que la difficulté de formuler des requêtes, même en NEXI, était rédhibitoire.

Quoi qu'il en soit, pour des utilisateurs occasionnels comme pour des experts, le langage de tous les jours ("naturel") semble être le moyen le plus simple et le plus intuitif d'exprimer un besoin d'information.

7.1.2 Connaissances de la structure

Outre leur apprentissage, les langages de requêtes formels requièrent de la part de l'utilisateur une très bonne connaissance de la structure des documents dans lesquels le système cherche les réponses aux questions posées. La sémantique de cette structure (des balises) doit également être maîtrisée.

Par exemple, pour retrouver une information provenant d'un résumé, d'une section, d'un élément bibliographique ou d'une figure, un utilisateur devra savoir si ces types d'éléments sont effectivement correctement identifiés dans la structure, et quel est le marquage (les noms de balises) correspondant (respectivement `'abs'`, `'sec'`, `'bb'` et `'fig'` dans les articles de l'IEEE d'INEX). Cette information est contenue dans la ou les DTD ou Schemas XML, mais il s'agit encore d'un nouveau langage à connaître, de plus de temps à consacrer et de nouvelles informations à retenir. Rappelons que la DTD des articles de l'IEEE comporte 192 éléments différents, celle de l'encyclopédie Wikipedia plusieurs centaines.

En outre, il existe des situations pour lesquelles le propriétaire de la collection ne

souhaite pas permettre l'accès à ces détails pour des raisons de respect de la vie privée, de maintenance ou de sécurité.

7.1.3 Les collections hétérogènes

collections hétéro-gènes

Dans le cas général, la recherche d'information doit se faire sur un ensemble de documents qui ne partagent pas la même structuration (qui possèdent des DTD ou des Schemas différents). On dit que la collection de recherche est "hétérogène". Même en considérant que les différentes structures sont relativement similaires (avec des unités de recherche semblables), les marquages utilisés sont très divers. Ainsi un paragraphe peut être isolé par des balises `'p'`, `'para'` ou `'paragraph'`. Des acronymes sont souvent utilisés (`'st'` pour *section title*), ainsi que des abréviations plus obscures (`'atl'` pour les titres dans IEEE).

Ainsi, une requête formelle doit être modifiée d'une structure à l'autre, et une recherche dans une collection hétérogène nécessite un nombre parfois considérable de formulations différentes. Une interface en langage naturel, autorisant l'utilisateur à exprimer son besoin de façon conceptuelle, peut permettre de résoudre ce problème, en analysant la phrase indépendamment du langage de requête et en effectuant ensuite une traduction en plusieurs expressions (une par DTD).

7.1.4 Une utilisation "intelligente" de la structure

Enfin, les documents utilisant une structure bien conçue et sémantiquement forte pour baliser le texte peuvent rendre la "compréhension" de la requête plus facile. De plus, les subtilités de la recherche dans les documents XML font que la requête d'un humain, même si elle est syntaxiquement correcte et sémantiquement adéquate, peut ne pas être formulée de la meilleure façon possible. Tout utilisateur habitué des moteurs de recherche (sur l'Internet par exemple) sait que ses requêtes, même composées de simples mots-clés, sont des adaptations de son besoin réel aux capacités du moteur qu'il utilise et qu'il a appris à connaître. Nous verrons que, dans le cas des documents XML, cette adaptation est difficile et qu'une expression trop naïve provoque des erreurs.

7.2 Ce que les ILN pour XML ne sont pas

7.2.1 Une technique de plus pour la recherche d'information

Comme nous l'avons vu (section 5), les techniques de traitement automatique de la langue adaptées à la recherche d'information ont été d'une efficacité relativement limitée. Les techniques fines sont finalement peu utilisées. Une des explications possibles est que les conditions habituelles de la RI traditionnelle font que les SRI n'ont pas besoin de "comprendre" la requête pour la traiter. En particulier, le fait que la recherche concerne de larges portions de texte permet d'utiliser des méthodes statistiques, qui s'avèrent plus efficaces que les approches linguistiques, elles-mêmes encore trop peu abouties.

A l'opposé, des domaines traitant des textes plus courts, comme par exemple le "question/réponse", font couramment usage du traitement de la langue. Ainsi nous pouvons espérer que le cas des documents XML apporte un nouveau champ d'étude plus heureux aux partisans des méthodes linguistiques.

7.2.2 Des interfaces pour les bases de données

BD vs. *RI*

Si quelques-unes des motivations évoquées ci-dessus sont communes avec le domaine des interfaces pour les bases de données, les problématiques restent bien distinctes :

- Nous avons déjà souligné que l'interrogation des bases de données est une interrogation *stricte* ; il ne s'agit pas de recherche d'information. L'utilisateur sait quel type de connaissance est stockée dans la base, son besoin est précis et une requête correcte conduit nécessairement à une réponse correcte. Une compréhension erronée par une ILN conduit à des résultats totalement inutiles – voire à aucun résultat. Ceci implique que l'analyse de la question en langage naturel doit interpréter le besoin de façon parfaite et non ambiguë, faute de quoi la réponse finale est incorrecte et l'utilisateur non satisfait. Pour cette raison les interfaces en langage naturel pour les bases de données ne s'appliquent qu'à des domaines restreints (comme les données géographiques, médicales, etc.) avec des langages également restreints (en fait pseudo-naturels).

  interrogation stricte

  A l'opposé, en recherche d'information semi-structurée, tout comme en RI classique, le besoin d'information est défini de façon vague, et il n'existe pas de réponse parfaite à la question. Une interface en langage naturel devient alors **un composant à part entière du processus de recherche**. Il peut notamment interpréter la requête de façon imparfaite mais retourner malgré tout des résultats utiles. D'un autre côté, il est tout à fait possible d'imaginer qu'une telle interface parvienne à renvoyer de *meilleurs* résultats que des requêtes manuelles (ce qui n'a aucun sens en base de données[1]).

  interrogation vague

- Qui plus est, les demandes de la recherche d'information n'entraînent, contrairement à l'interrogation des bases de données, aucune opération sur les données, comme les calculs mathématiques, les concaténations, l'agrégation, la restructuration, etc. La réponse est un ensemble d'éléments XML qui font partie de la collection initiale. **Cette différence modifie profondément le type de questions qui seront posées à un système**, et tout autant les approches choisies pour analyser les requêtes en langage naturel. Dans le cas des bases de données, on attend (et on exige) des phrases conventionnelles, exprimant un besoin finalement prévu à l'avance. L'enjeu pour le concepteur d'une interface est surtout de répertorier les constructions qu'un utilisateur sera susceptible d'employer pour parvenir à ses fins.

  En revanche, et paradoxalement, la relative simplicité conceptuelle des requêtes de RI donne une liberté accrue à l'utilisateur pour compliquer à souhait la forme de ses entrées. Seules les éventuelles allusions à la structure des documents permettent de cadrer la syntaxe employée. Les formes négatives sont volontiers exploitées pour préciser les éléments qu'il jugera non pertinents, et l'expression s'étend souvent sur plusieurs phrases. On part cependant de l'hypothèse, et nous reviendrons sur ce point, que les constructions utilisées sont généralement concises, sans ratiocination excessive, et que la personne réfléchit à son besoin avant de formuler sa requête. De plus, aucune recherche n'a encore porté sur la possibilité de dialoguer avec le système après l'obtention du résultat, et chaque requête est traitée de façon indépendante des précédentes.

Pour toutes ces raisons, développer des interfaces pour la recherche d'information XML est un domaine de recherche séparé, nécessitant ses propres solutions innovantes.

Enfin, contrairement aux bases de données, le format XML semble promis à une utilisation par le grand public, notamment au travers de l'Internet. Bien que des requêtes

[1] On n'aborde pas ici la question de l'efficacité des requêtes en termes de performances en temps de calcul ou en espace utilisé.

formelles, structurées, non ambiguës et lisibles par des machines soient indispensable pour supporter le processus de recherche, le besoin d'interfaces plus simples devrait devenir de plus en plus important dans le futur.

En retour, les ILN pour les collections XML doivent analyser toutes les requêtes, complexes ou non, syntaxiquement correctes ou non, écrites dans un langage naturel non restreint, même si cette analyse est partielle ou imparfaite. De plus on attend des applications plus générales et des systèmes indépendants du domaine.

7.2.3 Des systèmes de question/réponse

Les utilisateurs des systèmes de question/réponse posent des questions fermées, demandant une réponse courte et factuelle ("Quand Napoléon est-il mort ? → "en 1821"). Ces systèmes doivent sélectionner un segment de texte pertinent dans la collection. Dans le cas de la RI pour XML, l'unité de recherche, le doxel, bien que flexible, reste formellement délimitée par la structure du document ; de plus, le besoin d'information est souvent plus général et évoqué par des requêtes ouvertes.

Néanmoins, il existe certains types de requêtes concernant des éléments petits et précis (comme les dates de publications), et pour lesquelles les enjeux dans deux domaines sont similaires. De plus, la recherche d'éléments XML peut servir de recherche de passages efficace dans une phase préalable de pré-traitement en question/réponse.

7.3 La tâche "Langage Naturel" d'INEX

Le traitement automatique de la langue naturelle s'est invité pour la première fois à INEX lors de l'édition de 2004. L'unique consigne était alors de n'exploiter que le champ de *description* en langage naturel pour effectuer la recherche[1]. L'intermédiaire utilisé par la tâche *ad-hoc*, c'est-à-dire la requête NEXI écrite par l'auteur, était donc interdite aux participants de cette tâche. Ceux-ci utilisaient toutes les techniques de RI et de TAL qu'ils souhaitaient mettre en œuvre, et renvoyaient un ensemble ordonné de résultats (ou *run*), de la même façon que pour la tâche *ad-hoc*. Malgré un faible nombre de participants, les résultats se sont révélés encourageants, mais encore nettement en dessous des résultats des systèmes utilisant les requêtes formelles manuelles [45].

De plus, un biais théorique important empêchait une évaluation rigoureuse des systèmes. En effet, la comparaison des équipes participatrices concernait la sortie de leurs systèmes respectifs. La performance finale dépendait donc à la fois de la qualité de l'interface de traduction et de celle du moteur de recherche. Il était donc difficile d'interpréter les résultats.

En 2005, une nouvelle épreuve fut proposée. La possibilité fut offerte aux participants de réaliser une interface en langage naturel, transformant la partie *description* de la requête en une liste de mots-clés (*title*) et en un titre NEXI (*castitle*). Dans cette tâche, appelée NLQ2NEXI, aucun moteur de recherche n'était nécessaire. Les requêtes automatiquement générées par les participants étaient lancées sur un moteur de recherche commun fourni par les organisateurs (système $\mathcal{S}$ – voir la figure 34).

NLQ2NEXI

Dans cette configuration, le moteur est donc une donnée constante. On peut ainsi considérer que l'évaluation des résultats (*run i* sur la figure) est en fait, indirectement, une évaluation des requêtes. Celle-ci a alors deux aspects :

– Une comparaison des interfaces entre elles.

[1] Voir les détails concernant INEX, les requêtes, le langage NEXI, la tâche *ad-hoc* et son évaluation dans un autre rapport concernant la recherche d'information semi-structurée [97].

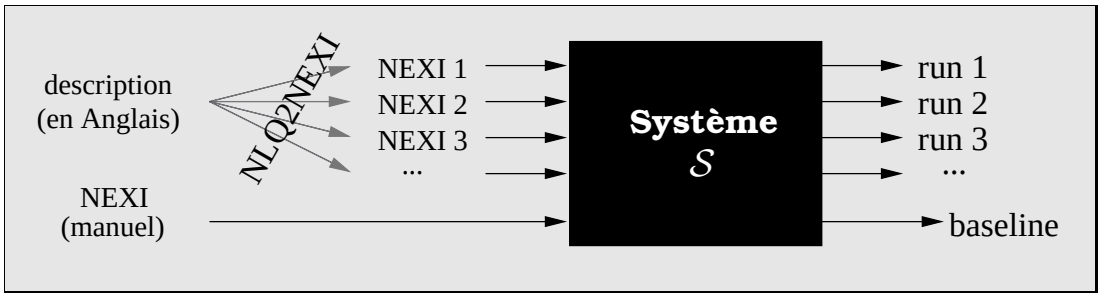


FIG. 34 – La tâche NLQ2NEXI.

– Une comparaison de chaque interface avec la requête manuelle, elle aussi lancée sur le moteur $\mathcal{S}$, aboutissant à un run "témoin", ou *baseline*.

Les mêmes mesures d'évaluation que pour la tâche *ad-hoc* étaient utilisées. Cette fois les résultats ont été très positifs, les performances des requêtes issues des interfaces en langage naturel se montrant comparables, voire meilleures dans certains cas que celles des requêtes manuelles.

7.4 Les premières approches

Nous présentons ici les techniques de *Woodley et Geva* [108] et de *Hassler*[1] visant à traduire les requêtes en langage naturel vers des expressions en NEXI.

Notre propre approche est détaillée notamment dans [96].

7.4.1 Pré-traitement

La première étape de la traduction est la distinction entre les constructions concernant la structure et celles concernant le contenu des documents recherchés.

Woodley et Geva effectuent une analyse morphosyntaxique de la requête avec Brill Tagger [18] et y incorporent principalement trois types d'informations supplémentaires à l'aide de dictionnaires spécifiques :

– les éléments structurels (*"structure"*, ou XST : références directes à des balises, comme *"abstract"* pour les résumés, *"figures"*, etc.) ;
– les indicateurs de relations (XBD pour *"boundaries"*), qui indiquent un lien entre deux éléments structurels ou entre un élément structurel et du contenu (comme les mots *"about"*, *"containing"*, etc., indépendamment de leur catégorie grammaticale) ;
– les instructions (XIN : *"find"*, *"retrieve"*) qui permettent de distinguer les éléments cibles* des éléments supports*.

Deux exemples d'étiquetage de requête sont donnés à la figure 35 [2]. Notons que les deux dernières propriétés (XBD et XIN) sont indépendantes de la collection de travail,

[1] Le travail de l'Université de Klagenfurt, par Marcus Hassler et Abdelhamid Bouchachia, n'a fait l'objet d'aucune publication jusqu'à présent. Ce que nous en savons nous vient de discussions informelles avec M. Hassler.

[2] Les exemples de cette section sont donnés en Anglais. En effet, les travaux décrits sont effectués dans le cadre d'INEX, avec des collections et des requêtes dans cette langue. Nos propres travaux ont été effectués sur l'Anglais. De plus, les catégories grammaticales utilisées désormais sont les catégories de la Penn Treebank, énumérées à l'annexe A.2.

The/DT relationship/NN and/CC comparisons/NNS between/IN radial/JJ basis/NNS functions/NNS and/CC multi/NN layer/NN perceptions/NNS

Find/XIN sections/XST about/XBD compression/NN in/IN articles/XST about/XBD algorithms/NNS

FIG. 35 – Etiquetage morphosyntaxique enrichi (*Woodley et Geva*).

```
Query: Request+
Request: CO_Request | CAS_Request
CO_Request: NounPhrase+
CAS_Request: SupportRequest | ReturnRequest
SupportRequest: Structure [Bound] NounPhrase+
ReturnRequest: Instruction Structure [Bound] NounPhrase+
```

FIG. 36 – Quelques patrons sémantiques présentés par *Woodley et Geva*.

tandis que la première (XST) demande des connaissances préalables et diffère selon la structure des documents.

Une analyse syntaxique superficielle est alors utilisée pour distinguer les éléments multi-termes (comme *figure caption* pour la légende d'une figure), mais aussi pour regrouper les syntagmes nominaux simples.

7.4.2 Analyse

Puis *Woodley et Geva* utilisent la technique des patrons sémantiques (décrite dans la section 4.2.4) pour faire le lien entre les éléments détectés. Ils partent de l'idée que des requêtes concernant la structure forment un contexte particulier pour lequel un nombre limité de patrons suffit [107] (voir figure 36). Ceux-ci permettent tout de même de détecter les constructions de base ainsi que certaines négations, la voix passive et quelques tournures anaphoriques [108].

Hassler emploie également des patrons sémantiques, mais basés directement sur les lemmes des termes de la requête et leurs catégories syntaxiques, obtenus avec TreeTagger [89]. Ceci est illustré par un exemple à la figure 37. Contrairement à l'autre approche présentée, les liens entre les éléments structurels et le contenu ne sont pas linguistiquement motivés ; il est supposé que le contenu textuel est inclu dans le dernier élément structurel introduit, ce qui est une simplification importante.

7.4.3 Formulation de la requête NEXI

La formulation des requêtes NEXI à partir de l'analyse sémantique est opérée de façon simple. L'élément cible est repéré par les termes d'instruction (XIN) chez *Woodley et Geva* ; *Hassler* décrète que la cible est le premier élément structurel de la phrase.

De façons très similaires pour les deux équipes, l'élément cible est placé à droite de la requête avec son contenu (comme le montre la figure 37). Ensuite, les parties du support sont disposées à gauche, en partant de l'ancêtre le plus éloigné.

Ainsi que le format de NEXI le permet, les syntagmes simples sont entourés par des guillemets.

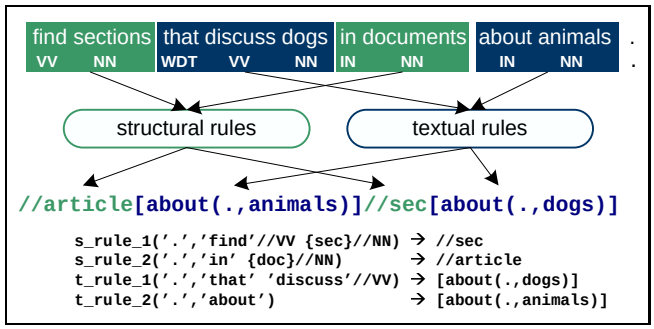


FIG. 37 – Analyse de la requête par *Hassler*.

7.4.4 Limites

La technique des patrons sémantiques est très efficace pour les requêtes exprimées de façon traditionnelle (c'est-à-dire les constructions prévues à l'avance). Elle évite les ambiguïtés syntaxiques mais manque de robustesse, puisqu'une phrase ne correspondant à aucun patron se verra analysée comme un sac de mots sans lien, ceci conduisant à de grandes imprécisions. En conséquence, l'approche de *Hassler*, qui ne comporte aucune phase syntaxique, est surtout performante dans la tâche *Content-Only* (CO), pour laquelle aucune référence à la structure n'est faite.

De plus, aucune des deux approches ne tient compte des allusions implicites à la structure, c'est-à-dire des formes qui introduisent une référence à une balise sans la nommer explicitement. Par exemple, dans l'exemple suivant, la construction *"documents écrits par"* (*"documents written by"*) introduit un auteur (balise `'author'`).

(56) We are looking for the documents written by Jiawey Han.
author

Prendre en compte de tels phénomènes demanderait un ajout de renseignements dépendants de la structure des documents, et donc une perte de généricité. S'en abstenir empêche pourtant de traiter bon nombre de tournures de phrases courantes.

Si les informations liées au type de corpus étudié semblent indispensables, au moins au niveau lexical (reconnaissance des termes se référant aux balises), personne n'a utilisé jusqu'à présent de connaissances concernant le domaine des documents (sur l'informatique par exemple pour les articles de l'IEEE). Le choix de l'encyclopédie Wikipedia [30], traitant des domaines extrêmement diversifiés, comme nouvelle collection en 2006 à INEX, n'incitera pas les chercheurs à s'engager dans cette voie.

Annexes

A Les catégories grammaticales

A.1 Aperçu rapide

On distingue les catégories fermées, dont les membres sont en nombre limité, des catégories ouvertes, qui évoluent et contiennent un nombre potentiellement infini d'éléments. Dans la première classe, on trouve :

préposition
déterminant
pronom
conjonction
auxiliaire

- les **prépositions** (*à*, *de*, *en*, *par*, *pour*, *sur*, etc.) ;
- les **déterminants** (*un*, *une*) ;
- les **pronoms** (*elle*, *qui*, *je*) ;
- les **conjonctions** (*et*, *mais*, *si*) ;
- les **verbes auxiliaires** (*être*, *avoir*) ;
- les **numéraux** (nombre infini mais connu) ;
- en Anglais, les **particules** (*up*, *down*, *out*).

Les classes ouvertes sont :

nom
verbe
adjectif
adverbe

- les **noms** communs et propres ;
- les **verbes** ;
- les **adjectifs** ;
- les **adverbes** (*délicatement*, *très*, *vite*).

A.2 Les étiquettes de la *Penn Treebank*

La *Penn Treebank* [66] est un corpus en Anglais d'environ 4,5 millions de mots. Ce corpus est annoté de façon semi-automatique par les catégories grammaticales des mots et comporte également des indications syntaxiques.

Le jeu d'étiquettes grammaticales utilisé pour ce projet est devenu un standard de la langue anglaise. Il comporte les 36 étiquettes que liste la table 1, ainsi que 12 étiquettes concernant la ponctuation. C'est cet ensemble qu'utilise pour l'Anglais l'analyseur morphosyntaxique TreeTagger [88] dont nous nous sommes servi, ainsi que l'analyseur syntaxique de surface Cass [4].

B Quelques éléments de grammaire

B.1 Les groupes nominaux

Les groupes nominaux sont organisés autour d'une *tête*, qui est le nom central. Les autres éléments sont des modificateurs.

Avant la tête, on trouve les déterminants (*"le"*, *"la"*, *"des"*, etc.) et les pré-déterminants (*"tout"*, *"tous"*). Les groupes adjectivaux peuvent être situés avant ou après (*"bleu"*, *"tout bleu"*, *"bleu comme une orange"*).

- le **chat**
- le premier et dernier **chat**
- tous les *chats*
- les deux **chats**
- les deux petits *chats*
- les *chats* blancs

Symbole	*Signification*
CC	conjonction de coordination
CD	nombre
DT	déterminant
EX	*there*
FW	mot étranger
IN	préposition or conjonction de subordination
JJ	adjectif
JJR	adjectif comparatif
JJS	adjectif superlatif
LS	marqueur de liste
MD	modal
NN	nom singulier
NNS	nom pluriel
NP	nom propre singulier
NPS	nom propre pluriel
PDT	prédéterminant
POS	marque du possessif (*'s*)
PP	pronom personnel
PP$	pronom possessif
RB	adverbe
RBR	adverbe comparatif
RBS	adverbe superlatif
RP	particule
SYM	symbol
TO	*to*
UH	interjection
VB	verbe, forme de base (infinitif)
VBD	verbe au passé
VBG	verbe au gérondif ou participe présent
VBN	participe passé
VBP	verbe au présent, sauf 3$^{\text{ème}}$ personne du singulier
VBZ	verbe au présent, 3$^{\text{ème}}$ personne du singulier
WDT	déterminant en '*wh*'
WP	pronom en '*wh*'
WP$	pronom possessif en '*wh*'
WRB	adverbe en '*wh*'

TAB. 1 – Les étiquettes de la *Penn Treebank*.

Après le nom de tête, on trouve des groupes prépositionnels (*le chat* ***sur le mur***), les clauses non définies (gérondifs, *le chat* ***faisant une petite sieste***) ou des propositions relatives (*le chat* ***qui ronronne au soleil***).

B.2 Les groupes verbaux

Les verbes peuvent être de trois types principaux : les verbes auxiliaires (*"être"* et *"avoir"* en Français), les verbes modaux (*"pouvoir"*, *"vouloir"*...) et les verbes principaux, qui sont tous les autres.

intransitif

transitif

On peut également distinguer les verbes selon leur transitivité. Les verbes *intransitifs* n'attendent aucun complément et peuvent faire du sens en eux-mêmes (*"Je cours"*, *"Il rit"*). Les verbes *transitifs* demandent un complément d'objet, direct pour les verbes transitifs directs (*"Il a retrouvé ses clés"*, *"Il a lu un livre"*), indirect pour les verbes transitifs indirects (*"J'obéis* à *ma mère"*, *"Je me méfie* de *lui"*). Certains verbes acceptent les deux types de compléments d'objet (*"Je donne ma bénédiction à leur union"*). Il faut remarquer que la plupart des verbes transitifs peuvent être utilisés de façon intransitive, avec d'autres types de compléments (*"Je lis dans mon lit"*, *"Je me méfie toujours"*).

Un groupe verbal est un verbe accompagné de ces compléments, d'objet ou autre.

B.3 Les propositions relatives

Les propositions relatives sont des groupes verbaux ou des phrases complètes que l'on attache à des noms ou des verbes pour les complémenter. Un pronom relatif, parfois optionnel (en Anglais), fait le lien entre les deux.

Exemples :

- La thèse que j'ai écrite.
- Le directeur qui m'a encadré.
- Je me demande qui a signé ce livre.

Ainsi, les groupes verbaux peuvent contenir des groupes nominaux, et réciproquement, ce qui conduit à une récursivité théoriquement infinie.

B.4 La phrase

Trois types essentiels de construction au niveau de la phrase peuvent être distingués.

Les phrases déclaratives. Ce sont les plus courantes, elles sont composées d'un groupe nominal sujet suivi d'un groupe verbal.

Par exemple :

- Le petit chat miaule
- Mon avion part à 23h.
- Les pâtes que nous avons mangées à Florence n'étaient pas aussi bonnes que celles que la cousine de Loïc nous a cuisinées hier soir.

Les phrases impératives. Elles commencent par un groupe verbal, n'ont pas de sujet et sont utilisées en général pour commander ou suggérer :

- Miaule, petit chat !
- Ferme la porte.
- Passe-moi le sel.

Les questions. On peut séparer les questions fermées (réponse oui ou non, les deux premiers exemples) et les questions ouvertes (avec un pronom interrogatif, les deux exemples suivants).

- Le chat miaule-t-il ?
- La période d'essai de deux ans va-t-elle être maintenue ?
- Pourquoi le chat miaule-t-il ?
- Combien de temps la période d'essai doit-elle durer ?

B.5 Les abréviations utilisées

Les abréviations utilisées pour représenter les constituants principaux sont les suivantes :

- **NP**, pour Noun Phrase (syntagme nominal) ;
- **VP**, pour Verbal Phrase (syntagme verbal) ;
- **PP**, pour Prepositional Phrase (syntagme prépositionnel) ;
- **S**, pour Sentence (phrase).

C Ambiguïtés syntaxiques

Les deux règles principales permettant de réduire les ambiguïtés syntaxiques sans apport sémantique sont celles de l'attachement minimal et de l'association droite.

La première stipule qu'en cas d'ambiguïté, les constructions arborescentes possédant le moins de nœuds doivent être préférées. Cette idée est illustrée à la figure 38, page 53, avec la phrase *"Les gendarmes interpellent un conducteur en état d'ivresse"*.

Quant à la règle de l'association droite, elle s'applique lorsque plusieurs constructions ont le même nombre de nœuds, et précise que les attachements de constituants doivent se faire avec le constituant le plus proche (le plus à droite). Voir la figure 39, page 54, pour la phrase *"Le ministre a dit qu'il ne démissionnerait pas aujourd'hui"*.

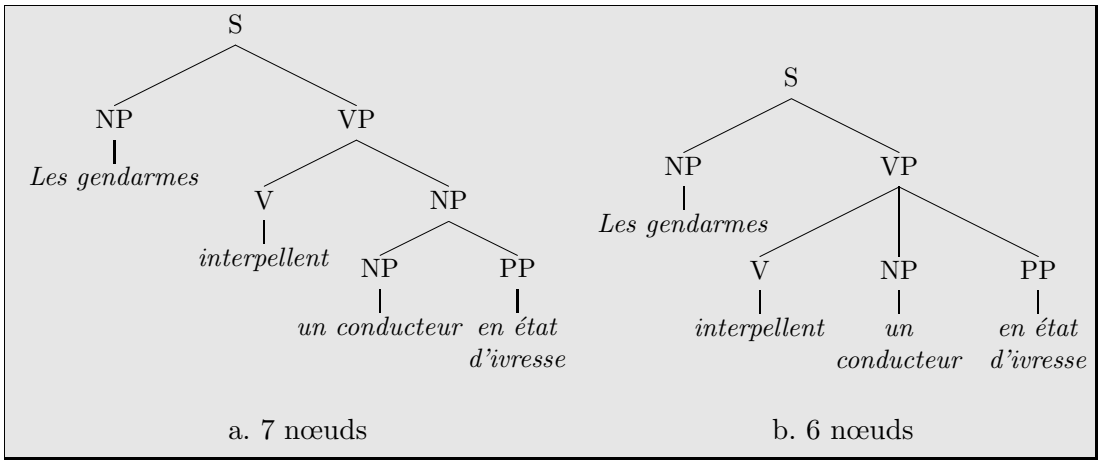


FIG. 38 – Règle de l'attachement minimal.

D L'analyse sémantique et le lambda-calcul

lambda-calcul

Le lambda-calcul (λ-calcul) constitue un formalisme capable de répondre aux problèmes de compositionnalité dans l'analyse sémantique [1]. Le λ-calcul est destiné à définir et à manipuler des fonctions [67] [2]. Les notions d'*abstraction* et d'*application* modélisent respectivement la *définition* et l'*appel* d'une fonction.

Ainsi, l'expression suivante est une abstraction :

(57) $\lambda x.chien(x)$

[1] Voir la section 4.2.1.

[2] Seul un très rapide aperçu du λ-calcul est donné ici, le lecteur intéressé pourra consulter les cours disponibles en ligne [13] ou d'autres ouvrages de référence [11].

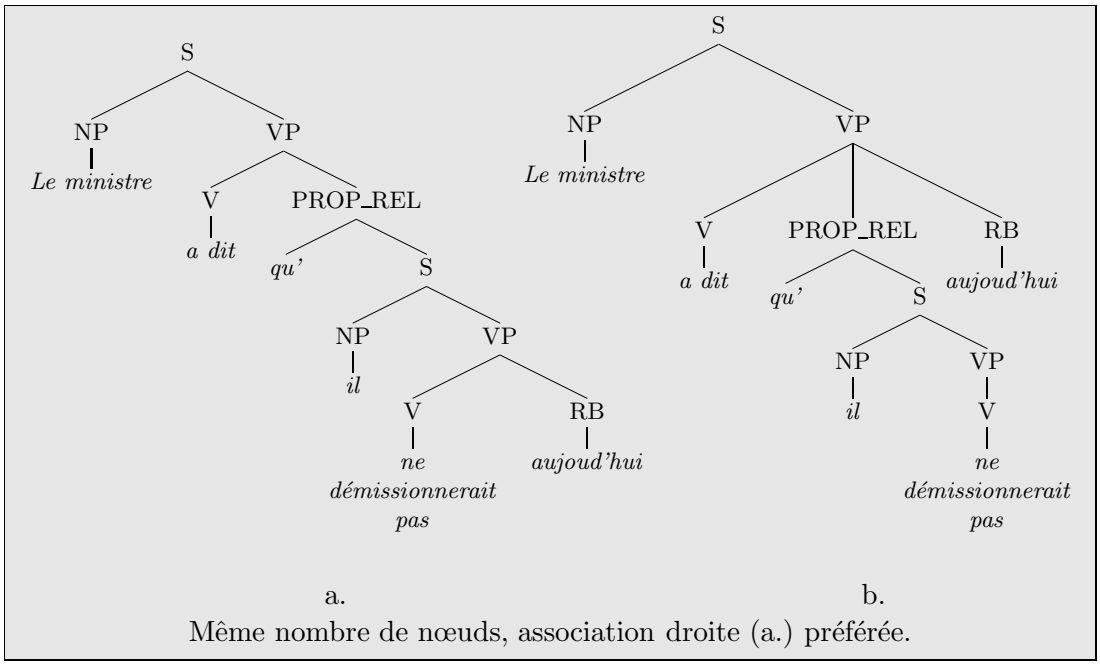


FIG. 39 – Règle de l'association droite.

Les *variables* accolées au λ sont les paramètres d'appel. Ici on peut appliquer la *constante* MILOU à cette application et obtenir, par un mécanisme de *réduction* (nous représentons l'application par un signe @) :

(58) $\lambda \boldsymbol{x}.chien(\boldsymbol{x})@\text{MILOU} \equiv chien(\text{MILOU})$

Ainsi chaque catégorie grammaticale peut être représentée par une abstraction. Par exemple :

- article indéterminé (*un*) : $\lambda P \lambda Q\ \exists x(P@x \wedge Q@x)$
- déterminant universel (*tous*) : $\lambda P \lambda Q\ \forall x(P@x \rightarrow Q@x)$
- un nom propre : $\lambda P.P@\text{MEDOR}$ (et pas simplement une constante comme dans l'exemple simplifié 58)
- verbe intransitif* (*miauler*) : $\lambda x.miauler(x)$
- verbe transitif direct* (*manger*) : $\lambda X.\lambda z.(X@(\lambda x.manger(z,x)))$

Et chaque règle syntaxique est l'occasion d'une application :

- Pour un syntagme nominal composé d'un article indéfini et d'un nom :

$$\overbrace{\lambda \boldsymbol{P} \lambda Q\ \exists x(\boldsymbol{P}@x \wedge Q@x)}^{un} \ @\ \overbrace{\underline{\lambda y.chat(y)}}^{chat} \equiv \lambda Q\ \exists x(\lambda \boldsymbol{y}.chat(\boldsymbol{y})@x \wedge Q@x)$$
$$\equiv$$
$$\equiv \boxed{\lambda Q\ \exists x(chat(x) \wedge Q@x)}$$

- Pour une phrase composée d'un syntagme nominal et d'un verbe intransitif :

$$\overbrace{\lambda \boldsymbol{Q}\ \exists x(chat(x) \wedge \boldsymbol{Q}@x)}^{un\ chat} \ @\ \overbrace{\underline{\lambda y.miauler(y)}}^{miaule} \equiv \exists x(chat(x) \wedge \lambda \boldsymbol{y}.miauler(\boldsymbol{y})@x)$$
$$\equiv$$
$$\equiv \boxed{\exists x(chat(x) \wedge miauler(x))}$$

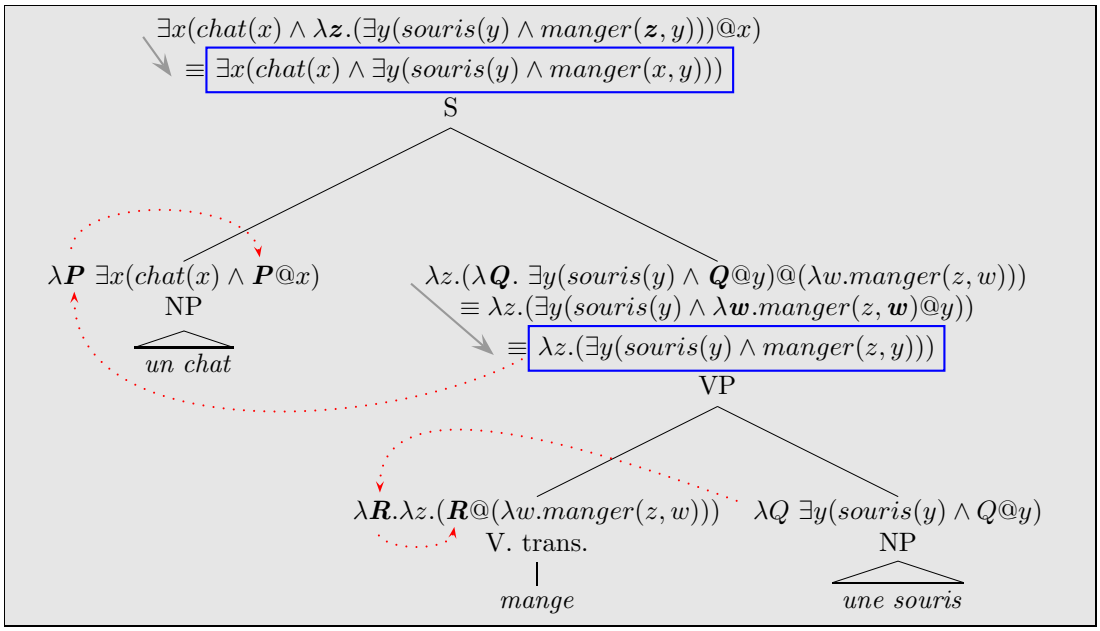


FIG. 40 – Analyse sémantique de la phrase *"Un chat mange une souris"*.

– On voit que l'application est la même si le syntagme nominal est un nom propre :

$$\overbrace{\lambda \boldsymbol{P}.\boldsymbol{P}@\text{TOM}}^{Tom} \ @\ \overbrace{\underline{\lambda y.miauler(y)}}^{miaule} \equiv \lambda \boldsymbol{y}.miauler(\boldsymbol{y})@\text{TOM}$$
$$\equiv$$
$$\equiv \boxed{miauler(\text{TOM})}$$

La figure 40 utilise ces exemples et illustre l'analyse sémantique d'une phrase complète.

Glossaire

Certaines des définitions ci-dessous sont partiellement ou totalement issues du *Dictionnaire de linguistique et des sciences du langage*, de Jean Dubois [34], ou du *Dictionnaire des sciences du langage* de Franck Neveu [75].

anaphore : Relation référentielle qui s'exerce à l'intérieur du discours entre deux expressions linguistiques, dont l'une reçoit son interprétation du sens référentiel de l'autre. Par exemple, par l'anaphore *pronominale*, un pronom est utilisé pour désigner un référent introduit au préalable. Voir la section 4.3.1 page 27.

association droite : Règle de réduction des ambiguïtés syntaxiques selon laquelle les attachements de constituants doivent se faire avec le constituant le plus proche (le plus à droite). Voir l'annexe C page 53.

attachement minimal : Règle de réduction des ambiguïtés syntaxiques selon laquelle les constructions arborescentes possédant le moins de nœuds doivent être préférées. Voir l'annexe C page 53.

composition : Procédé de création lexicale réalisée au moyen de la juxtaposition de plusieurs morphèmes* libres (*"malveillant"*, *"portemanteau"*, *"mâchefer"*...).

dérivation : Procédé qui consiste à former de nouveaux mots en modifiant le morphème par rapport à la base (par ajout d'un préfixe ou d'un suffixe.

ellipse : Suppression d'un constituant attendu dans le discours mais dont l'absence ne fait pas obstacle à l'interprétation de l'énoncé. Voir la section 4.3.2 page 29.

étiquetage morphosyntaxique : reconnaissance des mots et de leur catégorie grammaticale dans un texte.

flexion : Modification que subissent les mots qui se déclinent, se conjuguent, prennent la marque du pluriel, etc.

hyperonymie : On appelle *hyponomie* un lexème* *subordonné* à un autre lexème, qui lui est par conséquent *superordonné*, et qui est appelé *hyperonyme*. Par exemple, *"basset"*, *"pomme"*, *"tilleul"* sont des hyponymes de *"chien"*, *"fruit"*, *"arbre"*.

hyponymie : voir hyperonymie.

intransitif : un verbe intransitif est un verbe qui n'admet pas de complément d'objet.

langage naturel : Se dit du langage "normal" parlé par un être humain, quelle que soit sa langue. S'oppose aux langages de programmation formels en informatique.

lemme (1) (ou lexie) : unité autonome constituante du lexique d'une langue.

lemme (2) : racine d'un mot, dépouillée des marques d'accord et de conjugaison. C'est la forme graphique conventionnellement choisie comme adresse dans un lexique.

lexique : le lexique d'une langue constitue l'ensemble de ses lemmes* ou, d'une manière plus courante mais moins précise, "l'ensemble de ses mots".

morphème : unité minimale de signification (racine des mots).

morphologie (1) : étude de la forme des mots, à travers les phénomènes ressortissant de la flexion*, de la dérivation* et de la composition*.

morphologie (2) : étude conjointe des règles de structure interne des mots et de leurs règles de combinaison (morphosyntaxe).

polysémie : Propriété d'un mot d'avoir plusieurs significations distinctes.

pragmatique : étude de la signification des énoncés en lien avec le contexte (interlocuteurs, phrases précédentes, connaissance commune du monde, ...).

règle hors contexte : règle de grammaire de la forme G → D, dans laquelle G est un unique élément et D une séquence d'éléments, qui peut être remplacée par G, quel que soit leur contexte d'apparition (voir la section 3.3.2).

sémantique : étude de la signification des énoncés, indépendamment de tout contexte.

synonymie : Propriété de plusieurs mots d'avoir une même signification ou des significations approchées.

syntagme : Constituant syntaxique, suite de morphèmes* organisé autour d'une tête* et exerçant dans la phrase la même fonction syntaxique que celle-ci. Un syntagme nominal a pour tête un nom, un syntagme verbal, un verbe, etc.

syntaxe : partie de la grammaire décrivant les règles par lesquelles se combinent en phrases les unités significatives (mots).

tête : la tête est le constituant lexical principal du syntagme, dont la fonction et la distribution sont identiques à celles de l'ensemble du groupe.

trait : caractérique particulière d'un mot ou d'un constituant linguistique. Exemple : le genre, le nombre d'un nom, le temps d'un verbe.

transitif : un verbe transitif implique la présence d'un syntagme nominal* pour le compléter (complément d'objet). Par exemple, le verbe *aimer (quelqu'un)* est transitif.

Références

[1] A. Abeillé. *Les nouvelles syntaxes : grammaires d'unification et analyse du Français*. Collection Linguistique. Armand Colin, 1993. 20

[2] S. Abney. Part-of-Speech Tagging and Partial Parsing. In *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publisher, 1996. 13

[3] S. Abney. Partial Parsing via Finite-State Cascades. *Journal of Natural Language Engineering*, 2(4) :337–344, 1996. 14

[4] S. Abney. *The SCOL Manual*, Apr. 1997. http ://www.vinartus.net/spa/. 50

[5] J. Allen. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc, 1994. 5, 30, 34

[6] I. Androutsopoulos, G. Ritchie, and P. Thanisch. An Efficient and Portable Natural Language Query Interface for Relational Databases. In P. Chung, G. Lovegrove, and M. Ali, editors, *Proceedings of the 6th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, pages 327–330, Edimburgh, UK, June 1993. Gordon and Breach Publishers Inc., Langhorne, PA, USA. 40

[7] I. Androutsopoulos, G. Ritchie, and P. Thanisch. Natural Language Interfaces to Databases – An Introduction. *Journal of Natural Language Engineering*, 1(1) :29–81, 1995. 38, 42

[8] *Third Conference on Applied Natural Language Processing (ANLP-92)*, Trento, Italy, 1992. Association for Computational Linguistics, Morristown, NJ, USA. 59

[9] A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. Linguistically-motivated Information Retrieval. In A. Kent, editor, *Encyclopedia of Library and Information Science*, volume 69, pages 201–222. Marcel Dekker, Inc., New York, Basel, Dec. 2000. 35, 36

[10] M. Ballabriga. Sémantique du slogan publicitaire. In J.-M. Adam and M. Bonhomme, editors, *Analyse du discours publicitaire*, pages 95–112. Editions Universitaires du Sud, Toulouse, 2000. 31

[11] H. P. Barendregt. *The Lambda Calculus. Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, second edition, 1997. 53

[12] M. Bates and R. J. Bobrow. Information retrieval using a transportable natural language interface. In *SIGIR '83 : Proceedings of the 6th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–86, Bethesda, Maryland, USA, 1983. ACM Press, New York City, NY, USA. 40

[13] C. Berline. Cours de lambda-calcul (DEA), 2002. http ://www.pps.jussieu.fr/˜berline/Cours.html. 53

[14] P. Blache. Une introduction à HSPG, 1995. http ://www.lpl.univ-aix.fr/ blache/publis.html. 20

[15] P. Blackburn and J. Bos. *Representation and Inference for Natural Language ; A first Course in Computational Semantics*. ComSem, 1999. 17, 22, 25

[16] A. Bonnet. Les grammaires sémantiques, outil puissant pour interroger les bases de données en langage naturel. *R.A.I.R.O. Informatique*, 14(2) :137–148, 1980. 26

[17] D. Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 14th conference on Computational linguistics*,

pages 977–981, Nantes, France, 1992. Association for Computational Linguistics, Morristown, NJ, USA. 13

[18] E. Brill. A Simple Rule-Based Part of Speech Tagger. In ANLP92 [8], pages 152–155. 8, 35, 47

[19] R. Capindale and R. Crawford. Using a Natural Language Interface with Casual Users. *International Journal of Man-Machine Studies*, 32 :341–361, 1990. 42

[20] R. Carré, J.-F. Dégremont, M. Gross, J.-M. Pierrel, and G. Sabah. *Langage Humain et Machine*. Presse du CNRS, 1991. 36

[21] F. Carton. *Introduction à la phonétique du français*. Paris, Bordas, 1974. 4

[22] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143, Austin, Texas, USA, 1988. Association for Computational Linguistics, Morristown, NJ, USA. 8

[23] V. Claveau. *Acquisition automatique de lexiques sémantiques pour la recherche d'information*. PhD thesis, Université de Rennes 1, Dec. 2003. 36

[24] A. Copestake and D. Flickinger. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece, 2000. 13

[25] A. Copestake and K. S. Jones. Natural Language Interfaces to Databases. *The Knowledge Engineering Review*, 5(4) :225–249, 1990. 40, 41, 42

[26] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In ANLP92 [8], pages 133–140. 8

[27] S. Davis, editor. *Pragmatics, a Reader*. Oxford University Press, 1991. 5, 34

[28] R. S. de Madariaga, J. R. F. del Castillo, and J. R. Hilera. A Generalization of the Method for Evaluation of Stemming Algorithms Based on Error Counting. In M. Consens and G. Navarro, editors, *String Processing and Information Retrieval : 12th International Conference, SPIRE 2005*, volume 3772 of *Lecture Notes in Computer Science*, pages 228–233, Buenos Aires, Argentina, Nov. 2005. Springer-Verlag, New York City, NY, USA. 35

[29] S. M. Dekleva. Is Natural Language Querying Practical ? *SIGMIS Database*, 25(2) :24–36, 1994. 42

[30] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006. 49

[31] S. J. DeRose. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1) :31–39, 1988. 8

[32] A. Dister. Réflexions sur l'homographie et la désambiguïsation des formes les plus fréquentes. In *Actes des Journées Internationales d'Analyse Statistique des Données Textuelles (JADT 2000)*, 2000. 7

[33] H. Dreyfus. *What Computers Can't Do*. Harper Row, New York City, NY, USA, 1979. 21

[34] J. Dubois, editor. *Dictionnaire de linguistique et des sciences du langage*. Trésors du Français. Larousse, 1994. 56

[35] C. Fabre and C. Jacquemin. Boosting Variant Recognition with Light Semantics. In *Proceedings of the 18th International Conference on Computational Linguistics, COLING 2000*, pages 264–270, Saarbrücken, Aug. 2000. 35

[36] C. Fellbaum, editor. *WordNet : An Electronic Lexical Database*. The MIT Press, Cambridge, Massachussets, USA, 1998. 36

[37] I. O. for Standardisation (ISO). Information Technology – Database Language SQL. Standard No. ISO/IEC 9075 :1999. 36

[38] W. Francis. A tagged corpus – problems and prospects. In S. Greenbaum, G. Leech, and J. Svartvik, editors, *Studies in English linguistics for Randolph Quirk*, pages 192–209. Longman, 1979. 7

[39] C. Froissart. *Robustesse des interfaces homme-machine en langue naturelle*. PhD thesis, Université Pierre Mendès-France, Grenoble II, Dec. 1992. 13

[40] C. Fuchs, J. François, J.-F. L. Ny, and J.-L. Nespoulos. *La linguistique cognitive*. Ophrys, 2004. 5

[41] N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors. *Advances in XML Information Retrieval and Evaluation : Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*, volume 3493 of *Lecture Notes in Computer Science*, Schloss Dagstuhl, Germany, November 28-30, 2005, 2006. Springer-Verlag, New York City, NY, USA. 63

[42] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11) :964–971, Nov. 1987. 35

[43] C. Gardent and K. Baschung. *Techniques d'analyse et de génération pour la langue naturelle*. Adosa, 1994. 15

[44] E. Gaussier, C. Jacquemin, and P. Zweigenbaum. Traitement automatique des langues et recherche d'information. In E. Gaussier and M.-H. Stéphanini, editors, *Assistance intelligente à la recherche d'informations*. Hermes Sciences Publications, Lavoisier, Paris, 2003. 34

[45] S. Geva and T. Sahama. The NLP task at INEX 2004. *ACM SIGIR Forum*, 39(1) :50–53, 2005. 46

[46] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with WordNet synsets can improve Text Retrieval. In *Proceedings of the COLING/ACL'98 Workshop on Usage of WordNet for NLP*, pages 38–44, Montreal, Canada, 1998. 36

[47] H. P. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics*, volume 3, pages 41–58. Academic Press, New York City, NY, USA, 1975. 4

[48] G. G. Hendrix and J. G. Carbonell. A tutorial on natural-language processing. In *Proceedings of the ACM '81 conference*, pages 4–8. ACM Press, New York City, NY, USA, 1981. 36

[49] G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum. Developing a natural language interface to complex data. *ACM Transaction on Database Systems*, 3(2) :105–147, 1978. 30, 31, 39

[50] E. W. Hinrichs. Tense, Quantifiers, and Contexts. *Computational Linguistics*, 14(2) :3–14, 1988. 40

[51] J. R. Hobbs. Resolving Pronoun References. *Lingua*, 44 :311–338, 1978. 27, 29

[52] J. R. Hurford. *Semantics : a coursebook*. Cambridge University Press, 1983. 5

[53] T. Janssen. Compositionality. In van Benthem and ter Meulen [99], chapter 7. 23

[54] J.-H. Jayez. *Compréhension automatique du langage naturel – Le cas du goupe nominal en Français*. Masson, Paris, France, 1982. 33

[55] O. Jespersen. *A Modern English Grammar on Historical Principles, part VII : Syntax*. Allen and Unwin, London, 1954. 27

[56] D. Jurafsky and J. H. Martin. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2000. 8, 20, 22

[57] H. Kamp and U. Reyle. *From discourse to logic*. Kluwer Academic Publisher, 1993. 21

[58] A. H. Kao. Montague Grammar. Technical Report EECS 595, University of Michigan, 2004. 23

[59] F. Katamba. *An introduction to phonology*. Longman, 1989. 4

[60] M. Kay. Functional Unification Grammar : a formalism for machine translation. In *Proceedings of the 22nd annual meeting on Association for Computational Linguistics*, pages 75–78, Stanford, California, USA, 1984. Association for Computational Linguistics, Morristown, NJ, USA. 20

[61] G. Lallich-Boidin and D. Maret. *Recherche d'information et traitement de la langue – Fondements linguistiques et applications*. Les Cahiers de l'enssib. Presses de l'enssib, 2005. 34

[62] D. B. Lenat. CYC : A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11) :33–38, Nov. 1995. 34

[63] S. C. Levinson. *Pragmatics*. Cambridge University Press, 1983. 5, 34

[64] J. Lovins. Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, 11(1) :22–31, 1968. 34

[65] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. The MIT Press, Cambridge, Massachussets, USA, 1999. 6, 8

[66] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a Large Annotated Corpus of English : The Penn Treebank. *Computational Linguistics (Special Issue on Using Large Corpora)*, 19(2) :313–330, June 1993. 7, 50

[67] D. Marre. Programmation fonctionnelle. INSA Toulouse, 2000. Cours de Génie Industriel et Informatique. 53

[68] A. Martinet. Qu'est-ce que la morphologie ? *Cahiers Ferdinand de Saussure*, 26 :85–90, 1969. 6

[69] P. H. Matthews. *Morphology*. Cambridge University Press, 1991. 4

[70] G. A. Miller. WordNet : a lexical database for English. *Communications of the ACM*, 38(11) :39–41, Nov. 1995. 32, 36

[71] F. Moreau and V. Claveau. Extension de requêtes par relations morphologiques acquises automatiquement. In *Actes de la 3ème conférence en Recherche d'Information et Applications (CORIA'06)*, pages 181–204, Lyon, France, Mar. 2006. 34

[72] F. Moreau and P. Sébillot. Contributions des techniques du traitement automatique des langues à la recherche d'information. Technical Report 1690, IRISA, France, Feb. 2005. 35

[73] F. Namer. Flemm : Un analyseur Flexionnel du Français à base de règles. *Traitement automatique des langues pour la recherche d'information, numéro spécial de la revue T.A.L.*, 41(2) :523–548, 2000. 35

[74] F. Namer. Acquisition automatique de sens à partir d'opérations morphologiques en Français : études de cas. In *9ème Conférence annuelle de Traitement Automatique des Langues Naturelles*, pages 235–244, Nancy, France, June 2002. 9

[75] F. Neveu. *Dictionnaire des sciences du langage*. Armand Colin, 2004. 6, 56

[76] R. A. O'Keefe and A. Trotman. The Simplest Query Language That Could Possibly Work. In N. Fuhr, M. Lalmas, and S. Malik, editors, *Proceedings of the second Workshop of the Initiative for the Evaluation of XML retrieval (INEX), December 15–17, 2003*, pages 167–174, Schloss Dagstuhl, Germany, 2004. 43

[77] A. Pappa, G. Bernard, and H. Oukerradi. Détection automatique de frontières des phrases – Un système adaptatif multi-langues. In *Actes du congrès MAJECSTIC 2003*, Marseille, France, 2003. 6

[78] B. Partee. Montague Grammar. In van Benthem and ter Meulen [99], chapter 1. 23

[79] J.-M. Pierrel, editor. *Ingénierie des langues*. Hermes Sciences Publications, Lavoisier, Paris, 2000. 8

[80] M. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3) :130–137, 1980. 34

[81] P. Sabatier. *Contribution au développement d'interfaces en langage naturel*. PhD thesis, Université Paris VII, July 1987. 36

[82] P. Saint-Dizier and P. Muller. Fondements linguistiques et informatiques pour le traitement automatique de la langue naturelle écrite, 2004. Cours de Master 2. 10

[83] C. Samuelsson and A. Voutilainen. Comparing a linguistic and a stochastic tagger. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 246–253, Madrid, Spain, 1997. Association for Computational Linguistics, Morristown, NJ, USA. 8

[84] M. Sanderson. Word sense disambiguation and information retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 142–151, Dublin, Ireland, July 1994. Springer-Verlag, New York City, NY, USA. 36

[85] J. Savoy. Stemming of French words base on grammatical category. *Journal of American Society for Information Science*, 44(1) :1–9, 1993. 34

[86] J. Savoy. A stemming procedure and stopword list for general French corpora. *Journal of American Society for Information Science*, 50(10) :944–952, 1999. 34

[87] J. Savoy. Modèles en recherche d'information. In E. Gaussier and M.-H. Stéphanini, editors, *Assistance intelligente à la recherche d'informations*. Hermes Sciences Publications, Lavoisier, Paris, 2003. 34

[88] H. Schmid. TreeTagger - a language independent part-of-speech tagger. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html. 50

[89] H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Sept. 1994. 8, 35, 48

[90] C. L. Sidner. Focusing for interpretation of pronouns. *Computational Linguistics*, 7(4) :217–231, 1981. 29

[91] M. Silberztein. *Dictionnaires électroniques et analyse automatique de textes : le système INTEX*. Masson, Paris, 1993. 8

[92] D. Sleator and D. Temperley. Parsing English with a Link Grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, 1993. 13, 16

[93] A. F. Smeaton. Using NLP or NLP Resources for Information Retrieval Tasks. In Strzalkowski [94], pages 99–111. 36

[94] T. Strzalkowski, editor. *Natural Language Information Retrieval*. Kluwer Academic Publisher, Dordrecht, NL, 1999. 62, 63

[95] T. Strzalkowski, F. Lin, J. Wang, and J. Perz-Carballo. Evaluating Natural Language Processing Techniques in Information Retrieval. In Strzalkowski [94], pages 113–145. 35

[96] X. Tannier. From Natural Language to NEXI, an Interface for INEX 2005 Queries. In Fuhr et al. [41]. 47

[97] X. Tannier. Recherche d'information dans les documents semi-structurés. Technical report, Ecole Nationale Supérieure des Mines de Saint-Etienne, July 2006. 43, 46

[98] E. Tzoukermann, J. L. Klavans, and C. Jacquemin. Effective use of natural language processing techniques for automatic conflation of multi-word terms : the role of derivational morphology, part of speech tagging, and shallow parsing. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 148–155, Philadelphia, PA, USA, 1997. ACM Press, New York City, NY, USA. 34

[99] J. van Benthem and A. ter Meulen, editors. *Handbook of Logic and Language*. The MIT Press, Cambridge, Massachussets, USA, 1997. 60, 62

[100] R. van Zwol, J. Baas, H. van Oostendorp, and F. Wiering. Query Formulation for XML Retrieval with Bricks. In A. Trotman, M. Lalmas, and N. Fuhr, editors, *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 80–88, Glasgow, UK, Aug. 2005. 43

[101] K. A. Vanlehn. Determining the Scope of English Quantifiers. Technical Report AITR-483, MIT, Ma, USA, 1978. 31

[102] E. Voorhees. Using WordNet to Disambiguate Word Senses for Text Retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180, Pittsburgh, Pennsylvania, USA, 1993. ACM Press, New York City, NY, USA. 36

[103] E. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69, Dublin, Ireland, July 1994. Springer-Verlag, New York City, NY, USA. 36

[104] E. Voorhees. Using WordNet for Text Retrieval. In C. Fellbaum, editor, *WordNet : An Electronic Lexical Database*. The MIT Press, Cambridge, Massachussets, USA, 1998. 36

[105] XML Path Language (XPath). World Wide Web Consortium (W3C) Recommandation, 1999. http ://www.w3.org/TR/1999/REC-xpath-19991116. 43

[106] D. H. D. Warren and F. C. N. Pereira. An efficient easily adaptable system for interpreting natural language queries. *American Journal of Computational Linguistics*, 8(3-4) :110–122, 1982. 40

[107] A. Woodley and S. Geva. NLPX at INEX 2004. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlàvik, editors, *Advances in XML Information Retrieval. Third Workshop of the Initiative for the Evaluation of XML retrieval (INEX)*, volume 3493 of *Lecture Notes in Computer Science*, pages 382–394, Schloss Dagstuhl, Germany, December 6-8, 2004, 2005. Springer-Verlag, New York City, NY, USA. 48

[108] A. Woodley and S. Geva. NLPX at INEX 2005. In Fuhr et al. [41]. 47, 48

[109] W. Woods, R. Kaplan, and B. Webber. The Lunar Sciences Natural Language Information System : Final Report. Technical Report BBN Report 2378, Cambridge, MA, USA, 1972. 40

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

www.emse.fr