L'analyseur syntaxique Vergne-98 présenté aux actions d'évaluation GRACE et EASy

Jacques Vergne, Frédérick Houben

GREYC – Université de Caen
BP 5186 -14032 Caen cedex
{Jacques.Vergne, Frederick.Houben}@info.unicaen.fr
www.info.unicaen.fr/~jvergne

Mots-clés : analyseur syntaxique de phrases en français, analyseur déterministe, moteur à base de règles, constituants non récursifs

Keywords: French sentence syntactic parser, deterministic parser, rule based parser, non recursive constituents

Résumé Nous présentons courtement l'analyseur syntaxique Vergne-98, participant aux actions d'évaluation GRACE et EASy : les principaux concepts mis en œuvre, ainsi que les post-traitements nécessaires à l'action EASy.

Abstract We briefly present the syntactic parser Vergne-98, involved in the evaluation actions GRACE and EASy : main implemented concepts, and post-processing necessary to the EASy evaluation action.

Introduction

L'analyseur Vergne-98 est un analyseur syntaxique de phrases écrites en français; il est déterministe, et utilise des moteurs à base de règles; il est fondé sur une hiérarchie de constituants non récursifs : tokens, chunks et phrases. Il a été conçu et développé de 1985 à 1998. Il a d'abord été combinatoire[1] (démonstration au CoLing 1990 à Helsinki : Vergne, 1990), puis, à partir de 1993, il est devenu déterministe et de complexité pratique linéaire[2].

Cet analyseur est un logiciel d'étude, c'est-à-dire un moyen d'expérimenter et d'évaluer des concepts. Il a obtenu la meilleure évaluation à l'action GRACE (décision : 100%, précision : 94,5%), action d'évaluation des étiqueteurs du français, cette validation opératoire étant aussi une validation des concepts. On trouvera une présentation plus détaillée de cet analyseur dans notre mémoire d'Habilitation à Diriger des Recherches (Vergne, 1999, 3.3).

---

[1] Et donc de complexité théorique exponentielle (processus de parcours de l'arbre des catégories possibles des tokens), avec un nombre imprévisible de solutions (aucune ou beaucoup).

[2] Ce qui a été rendu possible par l'abandon des concepts de syntagme récursif, et de structures syntaxiques attendues représentées par une grammaire formelle (Vergne, 2001).

Il produit en sortie deux fichiers de résultats : les tokens étiquetés (sortie GRACE), et les chunks étiquetés et reliés par des relations de dépendance, de coordination, et d'antécédance des pronoms relatifs (sortie EASy). Ces deux fichiers utilisent les résultats complets de l'analyse (la mise en relation des chunks complète l'étiquetage des tokens), et chaque unité a une catégorie unique (toujours 100 % de décision, propriété d'un analyseur déterministe).

1 Principaux concepts

1.1 Hiérarchie de constituants non récursifs et stratégie d'analyse dans cette hiérarchie

Les constituants sont non récursifs, et forment une hiérarchie dont chaque niveau est typé : constituants physiques : texte, phrases, tokens, et constituants calculés : les chunks. On notera l'absence des concepts de proposition et de fonction dans la proposition (sinon implicitement dans les relations de dépendance sujet-verbe et verbe-objet).

1.1.1 Segmentation descendante dans la hiérarchie des constituants

Le texte est segmenté en phrases, par un traitement contextuel des points (point final de phrase, ou point d'abréviation). Puis les phrases sont segmentées en tokens. Un token peut être un mot, une ponctuation, un groupe de mots (locution, nombre composé), ou une partie de mot (les amalgames sont traités comme deux tokens : préposition+déterminant).

1.1.2 Analyse montante dans la hiérarchie des constituants

Les tokens sont étiquetés avec les ressources lexicales et des règles de déduction contextuelle (350 règles). Les chunks sont délimités et typés. La structure de la phrase est alors exprimée sous la forme de chunks nominaux ou verbaux, et de tokens externes aux chunks : conjonctions, pronoms relatifs, pronoms sujets, prépositions, adverbes de phrase, et ponctuations. C'est sur cette structure qu'est calculée la mise en relation des chunks.

1.2 Catégories de token et catégories de chunk

Le jeu des catégories de token est distributionnel : une catégorie regroupe les tokens d'une classe distributionnelle de tokens[3]. Une catégorie appartient soit au chunk nominal, soit au chunk verbal ou bien est externe au chunk; elle a une position à l'intérieur du chunk nominal ou verbal : en début ou fin de chunk, après ou avant telle autre classe. Le caractère distributionnel du jeu des catégories est la base sur laquelle repose la régularité des déductions contextuelles à l'intérieur d'un chunk nominal ou verbal. Ceci conduit à dissocier des catégories habituellement réunies. Par exemple, les adjectifs sont subdivisés en épithètes antéposées, postposées dans le chunk nominal, attributs dans le chunk verbal ; les «adjectifs» possessifs et démonstratifs sont inclus dans les déterminants, dans le chunk nominal.

1.3 Ressources

1.3.1 Ressources lexicales

Le lexique partiel (190 Ko) contient les mots grammaticaux, des adjectifs le plus souvent antéposés, les adverbes qui ne se terminent pas par *-ement*. Les radicaux verbaux sont dans un

[3] Pour des informations plus précises, cf. : www.info.unicaen.fr/~jvergne/cat_mots_SNR.html

fichier à part (45 Ko). Des règles sur les finales complètent les ressources lexicales (16 Ko) pour les noms, les adverbes en *-ement* et les néologies verbales (Vergne, 1999, 3.3.3).

1.3.2 Ressources syntaxiques

Les règles de syntaxe (120 Ko, environ 550 règles), sont interprétées par deux moteurs : le premier opère sur la structure en tokens, le deuxième sur la structure en chunks + tokens externes aux chunks (cf. 1.5 ci-dessous).

1.3.3 Collaboration entre ressources lexicales et ressources syntaxiques au niveau des tokens

Les ressources lexicales affectent une catégorie, le plus souvent unique par défaut pour chaque token. L'ensemble de catégories d'un token peut ensuite être modifié par des règles de déduction contextuelle. Ces règles sont affirmatives : dans tel contexte, ce token a telle(s) catégorie(s), ce qui permet d'étiqueter un token n'appartenant pas aux ressources lexicales, ou ayant localement une catégorie inhabituelle (Vergne, Giguet, 1988).

Chronologie de l'application des ressources lexicales et des paquets de règles syntaxiques :

- étiquetage par le lexique des mots grammaticaux;

- premier paquet : affectation d'une catégorie générique «token de chunk nominal» ou «token de chunk verbal» aux tokens suivant un mot grammatical étiqueté par le lexique (15 règles) ;

- étiquetage par les radicaux verbaux, et par les règles sur les finales («guesser»), par union avec les catégories posées précédemment ;

- deuxième paquet : suppression des polycatégories éventuelles par intersection des catégories posées par les règles avec les catégories posées précédemment ; propagation du non-désaccord genre-nombre dans le chunk nominal, et pose d'une frontière de chunk en cas de désaccord (environ 300 règles).

1.4 Mise en relation des chunks

La mise en relation des chunks (Vergne, 1999, 2.3.2) modélise une saturation de valence généralisée, en reprenant le concept de Tesnière, mais en l'appliquant non pas aux mots mais aux chunks, et en ne la limitant pas aux valences verbales, par généralisation à toute relation de dépendance ou de coordination entre deux chunks :

- premier temps : valence détectée et à saturer ; un premier chunk d'un type T1 donné est mis en attente d'un deuxième chunk d'un type T2 donné pour le type de relation caractérisé par la valence à saturer ;

- deuxième temps : saturation d'une valence ; arrivée du deuxième chunk de type T2, mise en relation des deux chunks, oubli de la valence maintenant saturée.

Ces deux temps de la mise en relation nécessitent donc deux règles pour chaque mise en relation. Les chunks en attente sont mémorisés dans une pile pour chaque type d'attente : un chunk nominal attend un chunk verbal (relier sujet-verbe), un chunk verbal attend un chunk nominal (relier verbe-sujet postposé ou relier verbe-objet), un chunk nominal attend un chunk nominal coordonné, ... (13 piles, 250 règles). Ce processus permet de relier deux chunks sans aucun attendu de structure sur ce qui les sépare; il fonctionne de la même manière qu'ils soient contigus ou éloignés. Tout processus de mise en relation P1 peut interagir avec un autre processus de mise en relation P2 par une action sur la pile de P2 (Vergne, 1999, 3.3.6) ; par exemple, un chunk sujet n'attend plus de coordonné après sa mise en relation avec son verbe.

1.5 Règles symboliques et moteurs à base de règles

Les règles sont de la forme : conditions => actions, portant des deux côtés sur un même nombre d'unités (tokens ou chunks). Les conditions portent sur les catégories, les attributs, les graphies, les lemmes verbaux, les relations (dépendance ou coordination), la présence dans une pile, un opérateur de non désaccord du token courant avec le token précédent (genre-nombre dans les chunks nominaux, personne-nombre dans les chunks verbaux), un opérateur de non désaccord du chunk courant avec un chunk empilé (personne-nombre dans la relation sujet-verbe), un opérateur d'isomorphisme entre chunks à coordonner. On utilise les opérateurs booléens (*non ou et*). L'utilisation du *non* permet d'écrire des règles exclusives les unes des autres, et donc moins sensibles à leur ordre d'évaluation. Les actions sont : affectation d'une valeur de catégorie, d'attribut, empiler ou dépiler un chunk, relier deux chunks. Pour chaque unité de la phrase (token ou chunk), chaque moteur passe les règles du paquet courant sur l'unité courante et son contexte, d'où une complexité théorique et pratique linéaire selon le nombre d'unités (Vergne, 1999, 3.3.5).

2 Quelques caractéristiques de l'implémentation

Cet analyseur est écrit en Pascal sur Mac OS, et est en cours de réécriture en java pour faciliter sa portabilité. Les sources font 1,4 Mo, et l'exécutable 460 Ko. Les ressources lexicales et syntaxiques font ensemble 370 Ko. L'analyseur complet fait donc 830 Ko au total.

3 Adaptation des sorties de l'analyseur au format d'entrée EASy

La sortie des chunks étiquetés et reliés, avec leurs tokens internes et les tokens externes, est sous la forme d'un fichier texte (cf. : www.info.unicaen.fr/~jvergne/format_prs.html). Ce format est ensuite transcodé en un fichier XML, sans aucune perte d'information (étiquettes, et valeur des attributs), qui, associé à des feuilles de style, permet des regards différents sur les résultats. Ce premier fichier XML constitue un format de description pivot, qui est ensuite projeté dans le format XML EASy. Cette projection n'est pas seulement un changement de notation : la segmentation en chunks est différente ; par exemple, l'adverbe postposé au verbe est inclus dans le chunk verbal dans notre format, alors que, dans le format EASY, il constitue à lui seul un chunk adverbial dépendant du chunk verbal. De plus, certaines informations ne doivent pas figurer dans le format EASY : catégorie et attributs des tokens, attributs de genre, nombre et personne des chunks.

Références

VERGNE J. (1990), A parser without a dictionary as a tool for research into French syntax, Actes de *CoLing 1990*, vol. 1, 70-72. (www.info.unicaen.fr/~jvergne/JVergneColing1990.pdf)

VERGNE J., GIGUET E. (1998), Regards Théoriques sur le "Tagging", Actes de *TALN 1998*, 22-31. (www.info.unicaen.fr/~jvergne/VergneGiguetTaln98.pdf)

VERGNE J. (1999), *Étude et modélisation de la syntaxe des langues à l'aide de l'ordinateur - Analyse syntaxique automatique non combinatoire*, Habilitation à Diriger des Recherches, Université de Caen. (www.info.unicaen.fr/~jvergne/HDR_J.Vergne.pdf)

VERGNE J. (2001), Analyse syntaxique automatique de langues : du combinatoire au calculatoire, Actes de *TALN 2001*, 15-29. (www.info.unicaen.fr/~jvergne/Taln2001FR_JV.pdf)