TALN 2005, Dourdan, 6-10 juin 2005

Premier bilan de la participation du LORIA à la campagne d'évaluation EASY

Azim Roussanaly , Benoît Crabbé , Jérôme Perrin

LORIA
BP 239, 54506 Vandoeuvre-lès-Nancy Cedex
{Azim.Roussanaly}|{Benoit.Crabbe}|{Jerome.Perrin}@loria.fr

Mots-clés : Analyseur syntaxique, évaluation
Keywords: Parser, evaluation

Résumé

Ce papier décrit LLP2, analyseur à base de grammaires d'arbres adjoints lexicalisés (LTAG) que nous avons utilisé pour le projet EASY (Évaluation des analyseurs syntaxiques) ainsi que les ressources linguistiques associées. Quelques commentaires à propos de cette expérience inspirés des premiers résultats obtenus, sont également présentés.

Abstract

This paper describes LLP2 the Lexicalized Tree Adjoining Grammar (LTAG) based parser we have used for the french evaluation project EASY.as well as associated linguistic resources. Moreover, some comments about this experiment based upon the first results are set out.

1 Analyseur LLP2

1.1 Caractéristiques

L'analyseur du LORIA utilisé pour la campagne EASY s'intitule LLP2. Il s'agit d'un analyseur de type *deep parser* qui s'appuie sur une grammaire d'arbres adjoints lexicalisés (LTAG) (Joshi et al 1975). L'algorithme implémenté est celui de l'analyse par connexité décrit dans (Lopez 1999). L'intégration d'un module de traitement de structures de traits et d'unification, permet de prendre en compte les traits *top* et *bottom* aux nœuds des LTAG. En d'autres termes, LLP2 a la capacité de traiter des *Feature-based* TAG (Vijay-Shanker et al.1988)

Cependant, la version actuelle ne permet pas de prendre en compte les arbres auxiliaires décrivant des adjonctions englobantes (*wrapping adjunction*). Par conséquent, formellement, l'analyse est restreinte aux grammaires d'arbres insérés (TIG) (Schabes et al 1995). LLP2 a été est développée en Java et est disponible sous licence GPL.

1.2 Architecture

LLP2 offre une boîte à outils constituée d'une bibliothèque logicielle et de divers utilitaires. Dans le cadre d'un traitement par lots d'un corpus de phrases, l'architecture est illustré à la Figure 1 :

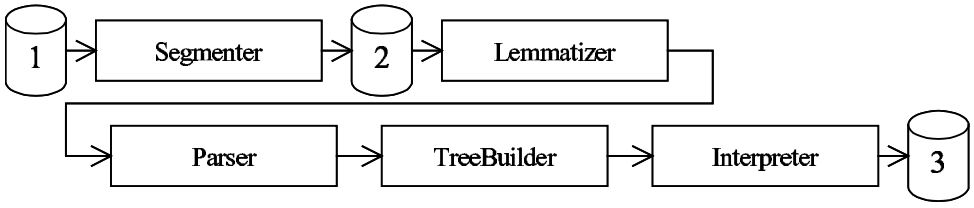

Figure 1 : Architecture LLP2

La liste de phrases à analyser est fournie sous forme d'un fichier texte standard (1). Le pré-processeur *Segmenter* effectue la tokenisation et l'analyse morphologique du texte initial. Le résultat intermédiaire est un liste d'unités lexicales étiquetées (que nous appelons segments) qui peut être stockée dans un fichier intermédiaire selon format XML que nous avons défini (2). Ce fichier est ensuite traité par le processeur *Lemmatizer* qui, en s'appuyant sur les lemmes identifiés lors du pré-traitement, se charge de relier les segments aux arbres élémentaires associés. L'étape suivante consiste à effectuer l'analyse syntaxique. Le résultat de cette étape est un état du *chart* à la fin de l'analyse. L'étape finale consiste à construire les arbres de dérivations et les arbres dérivés pour les analyses complètes et de les stocker dans un fichier résultat (3).

1.3 Adaptations pour la campagne EASY

Une première adaptation a été nécessaire en début de chaîne de traitement (voir Figure 2).

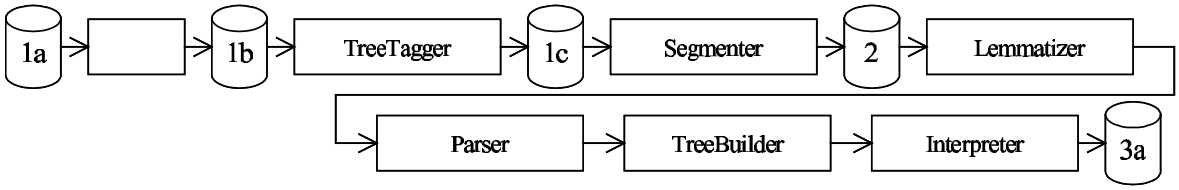

Figure 2 : Adaptation des entrées et des résultats

Ces adaptations sont motivées par :

- la nécessité de prendre en compte le format d'entrée imposé par EASY (textes déjà tokenisés). Il devient alors nécessaire de désactiver la fonction de tokenisation interne du processeur *Segmenter* afin d'éviter d'éventuels conflits avec la segmentation EASY et, de ce fait, de supprimer le risque de produire des résultats difficiles à synchroniser avec ceux attendus par EASY.
- l'usage conjoint d'un lexique morphologique très large et d'un lexique syntaxique peu contraignant (voir à la section suivante) provoquant la multiplication des arbres élémentaires à l'entrée et entraînant en définitive des ambiguïtés multiples et des temps d'analyse rédhibitoires

Ainsi le fichier en entrée devient celui fourni pour la campagne EASY (1a). Ce fichier est traduit dans un format d'entrée compatible avec *TreeTagger*.(Schmid 1994) (1b) dont l'usage permet de réduire les sources d'ambiguïtés. Le résultat de l'étiqueteur (1c) est ensuite

complété par le processeur *Segmenter* qui se contente d'enrichir les traits morphologiques et de remplacer éventuellement les unités inconnues. A ce stade du traitement, on obtient un fichier de segments étiquetés au format XML évoqué précédemment.

LLP2 fournit en résultat des arbres de dérivations et des arbres dérivés tandis que EASY attend un résultat sous la forme de relations de dépendance entre les segments présents dans une phrase. Un processeur (*Interpreter*) a été développé dans le but d'extraire les relations à partir des arbres de dérivations.

Par ailleurs, contrairement à une évaluation effectuée sur la base de TSNLP (Lehmann 1996), l'exploitation des analyses partielles, en cas d'échec d'analyse, permet de proposer certaines relations. Cela nous a conduit à développer un nouveau processeur *TreeBuilder* qui repose sur des heuristiques permettant de sélectionner les analyses partielles pertinentes.

2 Ressources

Du point de vue des ressources, LLP2 s'inspire de l'architecture XTAG (XTAG 1995, Crabbé 2004) qui distingue le lexique morphologique (permettant d'étiqueter les segments et d'identifier les lemmes correspondant), le lexique syntaxique (qui permet la sélection les arbres par filtrage et leur ancrage) et la grammaire (qui contient les arbres TAG).

2.1 Lexique morphologique

Pour la campagne EASY, le lexique morphologique est majoritairement construit à partir de MULTEXT (Ide 1994). Les principales modifications sont le fruit de l'adjonction de traits nécessaires à l'analyseur et à la mise en conformité des mots composés imposés par EASY.

2.2 Lexique syntaxique

C'est dans ce domaine que nous avons constaté les plus grandes lacunes de notre système en raison de l'absence de ressources syntaxiques réellement exploitables. Nous avons tout de même extrait un lexique syntaxique sur la base du lexique fourni par Lionel Clément et utilisé par l'analyseur XLFG (Clément 2001). Malgré quelques aménagements « manuels », cette ressource demeure encore incomplète. Un mécanisme par défaut de sélection des arbres élémentaires sur la base de règles reposant sur les traits morphologiques a dû être mis en place pour pallier les insuffisances du lexique syntaxique.

2.3 Grammaire

La grammaire que nous avons utilisée, a été engendrée à l'aide d'une méta-grammaire conçue par Benoît Crabbé (Crabbé 2005).et « compilé » avec à l'outil XMG développé au LORIA (Duchier et al. 2005). Une méta-grammaire peut être vue comme un moyen compacte d'exprimer une grammaire LTAG. Actuellement, la grammaire traite de manière satisfaisante les verbes et les adjectifs. Mais ce travail est encore en cours et la version utilisée pour la campagne comportait encore de nombreuses imperfections. Très récemment une évaluation avec TSNLP a été effectué avec des résultats encourageants. Des éléments chiffrés seront présentés lors de la session poster. Il aurait été judicieux de refaire les tests pour la campagne EASY avec cette nouvelle version.

3 Conclusion et perspectives

Il est indéniable que notre participation à EASY a été un moteur à nos travaux sur l'analyse syntaxique ; ce qui constitue en soi une expérience positive malgré le fait notre analyseur fournisse encore très peu de relations. Ce qui nous permet d'ores et déjà de penser que l'évaluation EASY sera sans aucun doute très négative. Mais nous pensons que ces résultats peuvent être nettement améliorés à court terme, en utilisant, d'une part, la dernière version de la grammaire qui a été testée sur le TSNLP et, d'autre part, en effectuant des « réglages » sur les stratégies de choix d'analyses partielles afin d'obtenir davantage de relations correctes lorsque l'analyse échoue. Cependant, il nous paraît impossible de parvenir à des résultats satisfaisants sans un effort de développement significatif au niveau du lexique syntaxique.

Nous considérons notre participation à la campagne EASY comme un point de référence de notre système. Nous espérons pouvoir réitérer régulièrement l'expérience afin de mesurer objectivement les améliorations des performances apportées par les solutions mise en œuvre dans le futur. Cette forme d'évaluation est complémentaire à une évaluation de type TSNLP.

Références

L. CLÉMENT: XLFG : A Parser to Learn LFG Framework, *NAACL 2001*, Pittsburgh

CRABBÉ, B, GAIFFE, B ET ROUSSANALY A. : Représentation et gestion de grammaires TAG *Revue TAL* , 2004

B. CRABBÉ :La représentation du lexique syntaxique, le cas de la grammaire d' arbres adjoints, *Thèse de doctorat Université Nancy2*, 2005 (à paraître)

D. DUCHIER, J. LE ROUX, Y. PARMENTIER : XMG, un compilateur de méta-grammaire extensible, *TALN 05* Dourdan, Juin 2005

N. IDE N., J. VÉRONIS:.MULTEXT (Multilingal Tools and Corpora*) COLING'94* Kyoto Japan 90-96, 1994

A. JOSHI, L. LEVI, M. TAKAHASHI : Tree Adjunct Grammars, *Journal of Compuper and System Sciences*, 1975

S. LEHMANN, D. ESTIVAL, S. OEPEN :N. TSNLP - Des jeux de phrases-test pour l'évaluation d'applications dans le domaine du TALN, *TALN* 96, Mai 1996 Marseille

P. LOPEZ : Analyse d'énoncés oraux pour le dialogue homme-machine à l'aide de grammaires lexicalisés d'arbres, *Thèse de doctorat UHP- Nancy1*, 1999

Y. SCHABES, R.C. WATERS : Tree Insertion Grammar : A Cubic Time Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced, *Computational Intelligence*, vol. 21, 479-514, 1995

H. SCHMID: Probabilistic Part-of-Speech Tagging Using Decision Trees, *International Conference on New Methods in Language Processing,* 1994

K. VIJAY-SHANKER, A. JOSHI: A Feature-based Tree Adjoining Grammar *COLING*'88, Budapest, 1988,

THE XTAG RESEARCH GROUP: A lexicalized tree adjoining grammar for English, *Technical Report IRCS Report 95-03*, The Institute for Research in Cognitive Science, Univ. of Pennsylvania, 1995