Segmentation en super-chunks

Olivier BLANC, Matthieu CONSTANT, Patrick WATRIN
IGM, Université de Marne-la-Vallée & CNRS
{oblanc,mconstan,watrin}@univ-mlv.fr

Résumé. Depuis l'analyseur développé par Harris à la fin des années 50, les unités polylexicales ont peu à peu été intégrées aux analyseurs syntaxiques. Cependant, pour la plupart, elles sont encore restreintes aux mots composés qui sont plus stables et moins nombreux. Toutefois, la langue est remplie d'expressions semi-figées qui forment également des unités sémantiques : les expressions adverbiales et les collocations. De même que pour les mots composés traditionnels, l'identification de ces structures limite la complexité combinatoire induite par l'ambiguïté lexicale. Dans cet article, nous détaillons une expérience qui intègre ces notions dans un processus de segmentation en super-chunks, préalable à l'analyse syntaxique. Nous montrons que notre chunker, développé pour le français, atteint une précision et un rappel de 92,9 % et 98,7 %, respectivement. Par ailleurs, les unités polylexicales réalisent 36,6 % des attachements internes aux constituants nominaux et prépositionnels.

Abstract. Since Harris' parser in the late 50's, multiword units have been progressively integrated in parsers. Nevertheless, in the most part, they are still restricted to compound words, that are more stable and less numerous. Actually, language is full of semi-frozen expressions that also form basic semantic units : semi-frozen adverbial expressions (*e.g.* time), collocations. Like compounds, the identification of these structures limits the combinatorial complexity induced by lexical ambiguity. In this paper, we detail an experiment that largely integrates these notions in a procedure of segmentation into super-chunks, preliminary to a parser. We show that the chunker, developped for French, reaches 92.9% precision and 98.7% recall. Moreover, multiword units realize 36.6% of the attachments within nominal and prepositional phrases.

Mots-clés : chunker, super-chunks, analyse syntaxique, patrons lexico-syntaxiques.

Keywords: chunker, super-chunks, syntactic analysis, lexico-syntactic patterns.

1 Introduction

Depuis l'analyseur syntaxique élaboré par l'équipe d'Harris à la fin des années 50 (Joshi & Hopely, 1997), les unités polylexicales ont progressivement été intégrées au processus d'analyse (Nivre & Nilsson, 2004). Cependant, dans la plupart des cas, elles sont restreintes aux mots composés, plus stables et moins nombreux. La langue regorge pourtant d'expressions moins figées qui peuvent également être considérées comme des unités sémantiques de base : les expressions adverbiales semi-figées et les collocations. De même que pour les composés, l'identification de ces structures facilite l'analyse syntaxique en limitant considérablement la combinatoire induite par l'ambiguïté lexicale.

Pour étudier ce phénomène, nous avons implémenté un *chunker*[1] reposant sur la notion de *super-chunk*. Ces structures diffèrent de la notion communément associée aux chunks (Abney, 1996; Karlsson *et al.*, 1995; Federici *et al.*, 1996; Ait-Mokhtar & Chanod, 1997) en ce qu'elles peuvent intégrer des attachements adjectivaux et/ou prépositionnels. Le super-chunk est donc une unité non récursive qui s'arrête à un élément lexical des classes N, V, A, Adv, ou à un élément complexe (MWU) appartenant à ces mêmes classes. Ainsi, par exemple, les séquences *chiffres d'affaires brut* et *marge d'exploitation*, étiquetées `N` (nom) lors de l'analyse lexicale, seront traîtées comme des mots simples durant la phase de segmentation[2]. Dans ce cas, la réduction de l'ambiguïté est évidente. Appréhendée de manière compositionnelle, la séquence *chiffres d'affaires brut* conduit à 24 analyses que nous linéarisons complètement si l'on envisage la collocation dans son ensemble. Par ailleurs, cette seule entrée lexicale nous permet de résoudre un double attachement (prépositionnel et adjectival), facilitant ainsi l'indentification des constituants.

Notre chunker s'inscrit dans un projet plus large visant l'analyse syntaxique du français. Telle que nous la concevons, cette analyse opère en trois phases de rafinement successifs : (1) la segmentation lexicale du texte en unités simples et complexes ; (2) la reconnaissance et l'étiquetage des super-chunks ; (3) l'attachement en constituants. Une illustration de cette procédure incrémentale est donnée au sein du tableau 1. Dans cet exposé, nous ne détaillerons pas plus en profondeur les caractéristiques de l'analyseur et limiterons notre propos à la segmentation en super-chunks. Nous nous concentrerons tout d'abord sur le module de segmentation lexicale en présentant les ressources utilisées. Nous montrerons comment une partie d'entre elles a été apprise automatiquement et comment nous les appliquons aux textes. Nous décrirons ensuite le module de segmentation en super-chunks inspiré par (Abney, 1996), et détaillerons la procédure de désambiguïsation. Enfin, nous évaluerons les performances de notre chunker et montrerons son intérêt pour la résolution d'attachements lexicaux.

NIVEAU	EXEMPLE
`Text`	Le groupe de télécommunications néerlandais KPN a annoncée avoir acquis une participation de 77,5 % dans le troisième opérateur allemand de téléphonie mobile E-Plus.
`Lexique`	Le [$_N$ groupe de télécommunications] néerlandais KPN a annoncé avoir acquis une participation de 77,5 % dans le troisième [$_N$ opérateur allemand de téléphonie mobile] E-Plus.
`Super-Chunk`	Le [$_N$ groupe de télécommunications] [$_{XA}$ néerlandais] KPN a annoncé [$_{XVI}$ avoir acquis] une participation de 77,5 % dans le [$_{XA}$ troisième] [$_N$ opérateur allemand de téléphonie mobile] E-Plus.
	[$_{XN}$ Le groupe de télécommunications] [$_{XA}$ néerlandais] [$_{XN}$ KPN] a annoncé [$_{XVI}$ avoir acquis] [$_{XN}$ une participation] de [$_{XN}$ 77,5 %] dans [$_{XN}$ le troisième opérateur allemand de téléphonie mobile E-Plus].
	[$_{XN}$ Le groupe de télécommunications] [$_{XA}$ néerlandais] [$_{XN}$ KPN] [$_{XV}$ a annoncé avoir acquis] [$_{XN}$ une participation] [$_{XP}$ de 77,5 %] [$_{XP}$ dans le troisième opérateur allemand de téléphonie mobile E-Plus].
`Syntagme`	[$_{N0}$ Le groupe de télécommunications néerlandais KPN] [$_V$ a annoncé avoir acquis] [$_{N1}$ une participation de 77,5 % dans le troisième opérateur allemand de téléphonie mobile E-Plus].

TAB. 1 – Processus global

[1]Les développements informatiques présentés dans ce travail reposent, en grande partie, sur la plate-forme logicielle Outilex (Blanc & Constant, 2006), développée à l'Université de Marne-la-Vallée (IGM).

[2]Notons que les informations morpho-syntaxiques sont héritées de la tête lexicale de l'unité complexe (*i.e. marge* et *chiffre*). De plus, nous associons à ces informations la structure interne de l'unité complexe (*i.e. nom-préposition-nom-adjectif* et *nom-préposition-nom*) afin de permettre une éventuelle décompression du tout (dans le but d'un étiquetage, par exemple).

2 Segmentation lexicale

Le processus de segmentation lexicale constitue la part fondamentale de notre chunker. Nous détaillons, dans cette section, les ressources utilisées de même que leur mode d'application.

Les ressources lexicales responsables de la segmentation se présentent sous deux formes : un ensemble de dictionnaires morpho-syntaxiques et une bibliothèque de grammaires locales. Ces ressources sont soit développées manuellement soit acquises automatiquement à partir de textes bruts.

2.1 Ressources lexicales construites manuellement

Les ressources lexicales développées manuellement s'organisent en un dictionnaire de formes fléchies (Courtois, 1990; Courtois *et al.*, 1997) et un réseau de 190 graphes ou grammaires locales[3].

Le dictionnaire compte 746 198 formes simples et 249 929 formes complexes (dont 245 436 noms[4]). Chaque entrée lexicale s'organise autour d'une forme fléchie, d'un lemme, d'une partie du discours, d'informations morphologiques (*e.g.* genre et nombre), d'informations syntaxiques (*e.g.* la structure interne des mots composés) et d'informations sémantiques (*e.g.* trait humain).

La bibliothèque de grammaires locales lexicalisées décrit un ensemble d'unités polylexicales[5]. Un exemple de grammaire locale est donnée à la figure 1. Cette grammaire décrit des adverbes de date et reconnaît des séquences comme *en mars 2007* et *cinq minutes plus tard*. Les chaînes entre < et > définissent des masques lexicaux[6] (*i.e.* les symboles terminaux). <*minute*>, par exemple, désigne les formes fléchies dont le lemme est *minute* (*i.e. minute* et *minutes*). Les sommets grisés sont, quant à eux, des références à d'autres graphes (*i.e.* les symboles non terminaux).

Notons que le graphe de la figure 1 définit un *transducteur* dont la sortie permet le balisage des séquences reconnues. Chaque adverbe de temps décrit par cette grammaire sera dès lors augmenté de l'étiquette `ADV+time`.

2.2 Collocations nominales et apprentissage

Outre les ressources lexicales développées manuellement, notre analyseur lexical intègre un ensemble de collocations nominales (*i.e.* des séquences de mots qui cooccurrent plus souvent qu'à la normale) apprises automatiquement. De cette manière, nous souhaitons favoriser la modularité de notre approche afin de la rendre viable dans un contexte applicatif réel tel que l'extraction d'information.

[3]Les grammaires locales sont des réseaux de transitions récursifs représentés sous la forme de graphes reconnaissant des langages algébriques (Gross, 1997; Woods, 1970). Elles permettent une représentation aisée des contraintes lexico-syntaxiques dans un contexte local.

[4]En marge des noms (*e.g. pomme de terre*, *faux témoignage*), il contient un ensemble de prépositions (*e.g. au milieu de*, *à cause de*), d'adverbes (*e.g. par ailleurs*, *en pratique*) et de conjonctions (*e.g. bien que*, *pendant que*)

[5]Des noms (*e.g. ministre anglais de l'Agriculture*), des prépositions (*e.g. à dix kilomètres au nord de*), des déterminants numériques (*e.g. vingt-sept*) et nominaux (*e.g. dix grammes de*) et des adverbes (*e.g. en octobre 2006*)

[6]Un masque lexical est une entrée lexicale sous-spécifiée équivalente à une structure de traits.

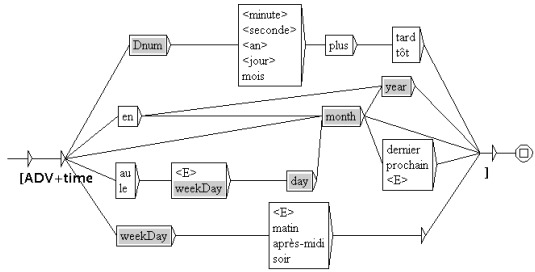

FIG. 1 – Une grammaire locale d'adverbes de date

Pour extraire les collocations, nous avons appliqué la méthode développée dans (Watrin, 2006) à un corpus de dépêches journalistiques d'un million de mots. Cette méthode, inspirée par (Daille, 1995), opère en trois étapes. Dans un premier temps, le corpus d'apprentissage est étiqueté, afin d'évacuer toute ambiguïté (principale source de bruit en extraction) et lemmatisé, pour permettre la généralisation des résultats. Ensuite, un ensemble de patrons syntaxiques, formalisant les structures de collocations, est appliqué au texte afin d'extraire les candidats termes. Finalement, les séquences identifiées sont évaluées statistiquement à l'aide du *log-likelihood* : (Dunning, 1993), pour les bigrammes et (Seretan *et al.*, 2003), pour les trigrammes.

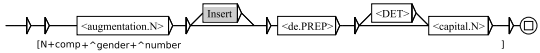

FIG. 2 – Collocation : *augmentation de capital*

Le processus d'extraction associe à chaque collocation sa structure interne. Cette structure nous permet de générer automatiquement les grammaires locales qui seront utilisées par le module de segmentation lexicale. Notons que ces grammaires locales prennent en compte d'éventuels modifieurs. Ainsi, par exemple, la grammaire associée à la collocation *augmentation de capital* (*cf.* FIG. 2) reconnaîtra la séquence *augmentations exceptionnelles de capital*.

L'extraction menée dans le cadre de cette expérience nous a permis d'isoler 1 953 formes canoniques (1 330 bigrammes et 163 trigrammes). Le nombre de collocations extraites pourrait paraître léger mais se justifie pleinement. Nous souhaitons automatiser au maximum le processus d'apprentissage tout en minimisant autant que possible le taux d'erreur. Par conséquent, nous utilisons des contraintes statistiques très fortes qui, si elles réduisent considérablement le nombre de collocation, assure la pertinence des résultats.

D'un point de vue pratique, nous avons observé que 69,1 % des bigrammes et 86,5 % des trigrammes extraits présentent une structure en *préposition-nom*. Ce constat appuie, selon nous, notre hypothèse d'un attachement au niveau lexical et justifie le repérage et l'étiquetage des collocations.

2.3 Application des ressources lexicales

Le module de segmentation lexicale se divise en deux étapes : (1) consultation du dictionnaire et (2) application des grammaires locales lexicalisées. Le programme de consultation du diction-

naire permet d'associer à chaque token toutes les étiquettes linguistiques potentielles et permet également de reconnaître et étiqueter les mots composés. La sortie de ce processus est un automate acyclique dans lequel chaque transition correspond à une entrée lexicale. De cette manière, nous pouvons conserver la totalité de l'ambiguïté. Les grammaires locales sont ensuite appliquées à cet automate, qui est alors augmenté des étiquettes associées aux unités polylexicales identifiées.

Bien que nous cherchions à conserver l'ambiguïté le plus loin possible dans notre chaîne de traitement, notre analyseur permet d'éviter certaines ambiguïtés *artificielles* en supprimant les analyses très rares du dictionnaire. Ainsi, par exemple, les analyses de *a* et *par* comme nom sont enlevées. Pour éviter le silence que peut provoquer la suppression de ces analyses, nous recourrons à un jeu des grammaires locales spécialisées formalisant de manière très précises leurs contextes d'apparition. Dès lors, la forme *par* sera toujours étiquetée *préposition*, sauf dans le cas où elle se trouve dans un contexte lexical particulier tel que *16 au-dessous du par* ou *faire le par*. Dans ce cas, elle sera également analysée comme nom.

3 Segmentation en super-chunks

La segmentation en super-chunks est également incrémentale. Elle consiste en une cascade de transducteurs appliqués à l'automate du texte. L'automate est ainsi augmenté à chaque étape des super-chunks identifiés. La cascade comporte huit étapes et utilise un réseau de 18 graphes reconnaissant successivement :

- les chunks adverbiaux (`XADV`) : les suites d'adverbes simples et les expressions adverbiales reconnues durant l'analyse lexicale ;
- les chunks adjectivaux (`XA`) : les suites d'adjectifs simples pouvant être précédées par un adverbe ;
- les chunks nominaux (`XN`) : les groupes nominaux simples, les entités nommées et certains types de pronoms ;
- les chunks prépositionnels (`XP`) : les `XN` précédés d'une préposition ;
- les chunks verbaux (cascade de 4 FSTs) : les voix actives et passives des infinitifs, participes passés, gérondifs et verbes conjugués (notés respectivement `XVI` – `XVIP` ; `XVK` – `XVKP` ; `XVG` – `XVGP` ; `XV` – `XVP`) ;

Les super-chunks héritent des propriétés morpho-syntaxiques de leur tête comme le montre la figure 3 qui représente un `XP`. `XP` hérite du lemme, du genre, du nombre et de la sous-catégorisation de sa tête (`^lemma`, `^gender`, `^number` et `^subcat`). Par ailleurs, nous conservons l'information liée à la préposition (`prep=$$.lemma`).

A la suite du processus de segmentation, l'automate du texte est nettoyé. La procédure de nettoyage consiste, d'une part, à supprimer les transitions dont les étiquettes n'appartiennent pas au niveau des super-chunks[7] (*e.g.* noms, verbes, adjectifs, ...) et, d'autre part, à conserver uniquement les chemins qui partent de l'état initial (début de phrase) et arrivent à l'état final (fin de phrase).

La procédure de segmentation en super-chunks appliquée à la séquence *au sujet d'un attentat terroriste* produit l'automate du texte donné à la figure 4.

[7] Notons toutefois que certaines entrées lexicales ne sont intégrées à aucun chunk (*i.e.* les conjonctions et les pronoms relatifs). Ces entrées sont conservées au même titre que les super-chunks.

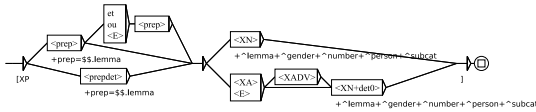

FIG. 3 – Chunk prépositionnel

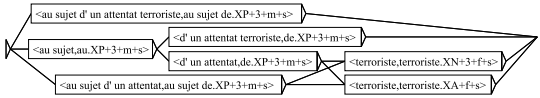

FIG. 4 – Automate du texte après segmentation

4 Levée d'ambiguïte incrémentale

La procédure de segmentation en super-chunks produit un ensemble d'analyses. Afin de réduire ou même de supprimer l'ambiguïté, le chunker inclut un module de levée d'ambiguïté composée de trois phases optionnelles : l'heuristique du plus court chemin, un jeu de règles et un module de décision statistique.

4.1 Application de l'heuristique du plus court chemin (SPH)

L'heuristique du plus court chemin consiste à ne garder, dans l'automate du texte, que les chemins les plus courts. Cette heuristique, indépendante de la langue, peut paraître simple et naïve au premier abord. Mais, en pratique, elle est très efficace. Elle privilégie en effet les analyses intégrant une ou plusieurs unités polylexicales au détriment des analyses compositionnelles. L'algorithme SPH est une adaptation de l'algorithme de Dijkstra (Dijkstra, 1959) qui garde l'ensemble des plus courts chemins d'un graphe au lieu d'un seul.

L'application de cette heuristique sur l'automate de la figure 4 du texte produit un automate totalement linéarisé : *<au sujet d'un attentat terroriste.XP>*.

4.2 Application de règles écrites manuellement

La plupart des ambiguïtés lexicales peuvent se résoudre efficacement en considérant leur contexte d'apparition. Dans cette perspective, nous avons développé un formalisme simple : les règles *Lubéron*. Une règle se compose de trois éléments : deux contextes (gauche et droit), éventuellement vides (`EMPTY`), représentés sous la forme de grammaires locales et une partie centrale listant une suite d'analyses ambiguës. Chaque règle décrit donc une ambiguïté potentielle[8]. Si cette dernière s'observe au sein de l'automate du texte, nous conservons uniquement la première analyse de la liste des éléments ambigus. Les autres analyses sont alors supprimées de l'automate.

`XN.wrtn`

[8]Le chunker contient actuellement 26 règles de ce type.

```
<XP> <XN>
EMPTY
```

La règle proposée ci-dessus exprime la contrainte suivante : dans le cas d'une ambiguïté XN – XP, l'analyse XP sera préférée si le contexte gauche (défini au sein du graphe XN.wrtn) présente un XN. Appliquée à l'automate de la figure 5, cette règle nous permet de supprimer l'analyse XN pour la séquence *de lutte contre le terrorisme*.

FIG. 5 – Ambiguïté des analyses en super-chunks

4.3 Application de règles statistiques simples

Certaines ambiguïtés ne peuvent être résolues efficacement par étude des contextes directs gauche ou droit. Un exemple prototypique est l'ambiguïté XV – XN (*e.g.* le mot *avions*, V (avoir) ou N (avion)). Dans ce cas, nous utilisons des règles de priorités statistiques apprises automatiquement au départ d'un corpus[9]. Étant donné un mot ambigu, l'analyse hors contexte la plus fréquente est choisie. Pour la forme *avions*, par exemple, nous retiendrons l'analyse N (probabilité de $0,6$) et supprimerons l'analyse V (probabilité de $0,4$). Si une forme ambiguë est absente de notre liste de décision, nous retenons, en dernier recours, la catégorie de (super-)chunk la plus fréquente.

Notons que toutes les phases de levée d'ambiguïté sont optionnelles. En effet, dans l'optique d'une analyse syntaxique, il peut être préférable de conserver une partie de cette ambiguïté, sa résolution pouvant entraîner des erreurs. L'ambiguïté XV – XN constitue, selon nous, une situation caractéristique qu'il est préférable de résoudre au niveau de l'attachement syntagmatique surtout si celui-ci est basé sur des règles lexicales.

5 Évaluation et discussion

La notion de super-chunk compatible avec notre définition n'existant dans aucun corpus annoté de référence, l'évaluation a dû être réalisée manuellement.

Notre procédure d'évaluation a porté sur un corpus composé de dépêches journalistiques extraites du site yahoo.fr. Ce corpus de 13 493 mots (*i.e.* 6 901 chunks), auxquels nous avons appliqué notre chunker à l'aide des données lexicales décrites dans les sections précédentes. La sortie est un texte annoté ne contenant plus aucune ambiguïté. Les résultats de l'évaluation sont donnés dans la table 2.

De manière générale, nous avons observé que la plupart des erreurs sont dues à l'incomplétude de nos ressources lexico-syntaxiques. Ceci implique que de nombreuses améliorations peuvent être apportées facilement et le seront très prochainement. Dans cette évaluation, nous distinguons les erreurs de précision et de rappel.

[9]Notre corpus comprend un an d'articles du journal Le Monde et a été étiqueté avec *TreeTagger* (Schmid, 1994).

PRÉCISION	RAPPEL	F MESURE
92,91 %	98,69 %	95.71%

TAB. 2 – Résultats

Les erreurs de **rappel** sont uniquement dues à la couverture de notre dictionnaire et de nos grammaires locales lexicalisées. D'un point de vue lexical, certains mots grammaticaux composés sont absents du dictionnaire (*e.g. tandis que*, *au-dessous de*). D'un point de vue plus grammaticale, les erreurs sont principalement imputables à la formalisation des entités nommées. La séquence *Nouri al Maliki*, par exemple, n'a pu être reconnue : la forme *al* est inconnue et n'a pas été intégré dans la grammaire comme un préfixe de nom de famille. Par ailleurs, quelques expressions semi-figées ont été oubliées (*e.g. vers 8h45*) de même que certaines structures pronominales complexes (*e.g. au cours de laquelle*).

Les erreurs de **précision** peuvent être divisées en quatre classes.

1. Erreurs liées à SPH

L'ambiguïté lexicale peut conduire a une mauvaise limitation des chunks après application de l'heuristique des plus courts chemins. Par exemple, dans la séquence *après l'affirmation du quotidien espagnol El Pais*, il existe deux analyses possibles :

– *[après l'affirmation XP] [du quotidien espagnol XP] [El Pais XN]* ;
– *[après l'affirmation XP] [du quotidien XP] [espagnol XA] [El Pais XN].*

Comme *quotidien* et *espagnol* peuvent être tous deux soit adjectif soit nom, l'algorithme SPH va préférer l'analyse `[Prep XA N]` au lieu de `[Prep N] [XA]`.

2. Erreurs dues aux règles statistiques

Dans la séquence *La côte Est et les villes de New York ...*, deux analyses sont attribuées au chunk *Est* : il s'agit soit d'un `XV` (être), soit d'un `XA` (direction est). Bien qu'*Est* soit `XA` dans ce contexte, le module statistique va préférer l'analyse `XV` (probabilité de $0,9$ contre $0,1$ pour l'analyse *XA*).

3. Erreurs causées par l'application des règles Luberon

Ces erreurs sont heureusement très rares. Elles concernent principalement l'ambiguïté `XP–XN` due à la forme *de* qui peut être déterminant et préposition. Dans la séquence *qui n'a pas fourni de plus amples détails*, par exemple, le chunk *de plus amples détails* aurait dû être étiqueté *XN*.

4. Erreurs imputables à la couverture lexicale

Quelques structures composées absentes dans le dictionnaire provoque des erreurs. Par exemple, *en outre* est un adverbe composé mais est absent de notre dictionnaire. Ainsi, l'analyse compositionnelle est choisie dans la phrase *ils ont en outre pris plusieurs centaines de personnes en otage*. Elle est segmentée en super-chunks de la manière suivante :

– *[ils XN] [ont XV] [en outre pris plusieurs centaines XP] [de personnes XP] [en otage XP]*

au lieu de,

– *[ils XN] [ont en outre pris XV] [plusieurs centaines XN] [de personnes XP] [en otage XP]*

où *en outre* est un adverbe inséré dans un chunk verbal.

En plus de l'évaluation en rappel et précision, nous avons aussi estimé l'impact des unités polylexicales pour l'attachement lexical. Notre procédure permet la réalisation correcte de 36,6 % des attachements lexicaux intérieurs aux groupes nominaux et prépositionnels, soit environ 13 % des attachements internes et externes aux syntagmes.

Malgré ces quelques erreurs, notre évaluation montre, selon nous, l'intérêt d'une segmentation en super-chunks, tant du point de vue de l'attachement que du point de vue de la réduction globale de l'ambiguïté.

6 Conclusion et perspectives

Dans cette article, nous avons présenté une technique de *chunking* reposant sur une augmentation significative du niveau lexical. En introduisant la notion de *super-chunks*, nous cherchions, d'une part, à optimiser le processus de désambiguïsation et, d'autre part, à résoudre une part de l'attachement lexical au sein des constituants prépositionnels et nominaux.

Afin d'évaluer la pertinence et l'efficacité de notre hypothèse, nous avons confronté notre chunker à un corpus de dépêches journalistiques. Cette expérience nous a permis de dégager une double conclusion.

– Notre procédure affiche une précision et un rappel excellents sans nécessiter le recours à un étiqueteur.
– La prise en compte des unités polylexicales nous permet d'évacuer efficacement (*i.e.* sans entraîner d'erreurs) une part conséquente des attachements internes aux constituants nominaux et prépositionnels.

Cette expérience nous a également permis de préciser un certain nombre de perspectives organisant notre travail futur. Ces perspectives s'articulent autour de deux points principaux : (1) l'augmentation des ressources lexicales (principal facteur de succès de notre application) et (2) l'amélioration du module de désambiguïsation statistique (en ce sens, l'intégration des HMM nous semble être une solution intéressante).

Références

ABNEY S. P. (1996). Partial parsing via finite-state cascades. *Natural Language Engineering*, **2**(4), 337–344.

AIT-MOKHTAR S. & CHANOD J.-P. (1997). Incremental finite-state parsing. In *Proceedings of the fifth Conference on Applied Natural Language Processing ANLP'97*.

BLANC O. & CONSTANT M. (2006). Outilex, a linguistic platform for text processing. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, p. 73–76.

COURTOIS B. (1990). Un système de dictionnaires électroniques pour les mots simples du français. *Langue Française*, **87**, 11–22.

COURTOIS B., GARRIGUES M., GROSS G., GROSS M., JUNG R., MATHIEU-COLAS M., MONCEAUX A., PONCET-MONTANGE A., SILBERZTEIN M. & VIVÈS R. (1997). *Dictionnaire électronique DELAC : les mots composés binaires*. Rapport interne, LADL (Paris 7).

DAILLE B. (1995). *Combined approach for terminology extraction : lexical statistics and linguistic filtering*. Rapport interne, Lancaster University.

DIJKSTRA E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271.

DUNNING T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, **19**(1), 61–74.

FEDERICI S., MONTEMAGNI S. & PIRELLI V. (1996). Shallow parsing and text chunking : A view on underspecification in syntax. In *Proceedings of the ESSLLI'96 Workshop on Robust Parsing*.

GROSS M. (1997). *The construction of local grammars*, p. 329–352. MIT Press : Cambridge.

JOSHI A. & HOPELY P. (1997). A parser from antiquity : an early application of finite state transducers to natural language parsing. *Natural Language Engineering*, **2**(4), 6–15.

KARLSSON F., VOUTILAINEN A., HEIKKILA J. & ANTTILA A. (1995). *Constraint Grammar : A language-independent system for parsing unrestricted text*, volume 4 of *Natural Language Processing*. Mouton de Gruyter.

NIVRE J. & NILSSON J. (2004). Multiword units in syntactic parsing. In *Workshop on Methodologies and Evaluation of Multiword Units in Real-World Applications*, p. 39–46, Lisbon.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

SERETAN V., NERIMA L. & WEHRLI E. (2003). Extraction of multi-word collocations using syntactic bigram composition. In *Proceedings of th* 4^{th} *International Conference on Recent Advances in NLP (RANLP-2003)*, p. 424–431.

WATRIN P. (2006). *Une approche hybride de l'extraction d'information : sous-langages et lexique-grammaire*. PhD thesis, Université catholique de Louvain.

WOODS W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, **13**(10).