Mettre les expressions multi-mots au coeur de l'analyse automatique de textes : sur l'exploitation de ressources symboliques externes

Mathieu Constant

Université Paris-Est
Habilitation à Diriger des Recherches en informatique

Mettre les expressions multi-mots au coeur de l'analyse automatique de textes : sur l'exploitation de ressources symboliques externes

Matthieu Constant

Soutenu le lundi 3 décembre 2012.
Le jury était constitué de

Prof. Béatrice Daille (examinatrice)
Prof. Laurence Danlos (présidente)
Prof. Cédrick Fairon (examinateur)
Prof. Éric Laporte (examinateur)
Prof. Alexis Nasr (rapporteur)
Prof. Éric Wehrli (rapporteur)
Prof. François Yvon (rapporteur)

Table des matières

Avant-propos

Le présent document synthétise et met en perspective mes travaux de recherche depuis mes débuts en thèse en 2000. J'insisterai plus particulièrement sur mes recherches post-thèse de 2003 à aujourd'hui. Mon domaine de recherche est le Traitement Automatique des Langues (TAL). L'un de mes objectifs principaux a été d'améliorer la finesse linguistique de différents analyseurs et diverses applications du TAL, en prenant en compte des phénomènes linguistiques particuliers, les expressions multi-mots : soit au moyen de prétraitements fins dans des applications ; soit en les intégrant dans divers modèles d'analyse. Mon idée directrice a été d'exploiter des ressources lexicales riches et de les coupler à différents modèles probabilistes ou procédures hybrides.

Bien que mon environnement quotidien de recherche (Laboratoire d'Informatique Gaspard-Monge – LIGM) n'a pas vraiment changé à l'exception d'une année en tant que chercheur dans une entreprise aux Etats-Unis, j'ai tenté de faire preuve d'une constante ouverture scientifique. Je suis venu au monde (de la recherche) dans le cadre du lexique-grammaire une méthodologie mise au point par M. Gross, cherchant à décrire de manière systématique les différents phénomènes de la langue. C'est donc naturellement que j'ai travaillé sur l'étude de phénomènes linguistiques particuliers : expressions de mesure et groupes prépositionnels géographiques. Ces expressions sont liées aux entités nommées qui au moment de ma thèse commençaient à poindre sérieusement le bout de leur nez dans la communauté du TAL. Ces études ont conduit à l'élaboration de ressources lexicales sous forme de lexiques tabulaires et grammaires locales, ainsi qu'au développement d'outils pour reconnaître ces expressions dans les textes.

Mon année post-doctorale (2003-2004) à Teragram Corporation (Boston, États-Unis) sous la direction d'E. Roche et Y. Schabes m'a ouvert aux techniques statistiques des sacs de mots et aux algorithmes sur les graphes de similarité de mots. J'ai, entre autres, mis au point un système inspiré du PageRank de Google afin d'extraire automatiquement les mots-clés (simples ou complexes) de documents textuels. Ces techniques basées sur les graphes de mots ont depuis montré de nombreuses applications comme le résumé automatique. A mon retour en France, j'ai continué à garder des liens avec le monde industriel et à développer des applications hybrides combinant approches statistiques et prétraitements linguistiques fins à base de ressources lexicales.

- Projet *Outilex* (Septembre 2005 - Octobre 2006) : finalisation d'une plateforme de traitement automatique de textes impliquant plusieurs partenaires industriels (avec O. Blanc)
- Entreprise *Softissimo* (juillet/septembre 2005) : extraction d'entités nommées dans un corpus tout en majuscule
- Entreprise *SeniorPlanet* (Janvier - Novembre 2006) : extraction de mots clés dans des articles du journal en-ligne SeniorPlanet (avec E. Laporte et S. Paumier)
- Entreprise *Xeres* (Avril - Juillet 2008) : veille sur le Web et regroupement de pages web par thématique (encadrement du stage de master 1

d'Anthony Sigogne)

Après mon recrutement en tant que Maître de Conférences en 2006, je me suis intéressé à l'hybridation des analyseurs linguistiques à travers l'exploitation de ressources lexicales fines et l'intégration de la reconnaissance d'expressions multi-mots : tout d'abord, avec des approches plutôt symboliques, puis au moyen d'approches probabilistes avancées. Toutes nos études ont été expérimentées sur des données écrites, et un certain nombre d'entre elles ont été appliquées sur des transcriptions orales.

Ces récentes évolutions de recherche ont, en partie, été rendues possibles avec le co-encadrement de trois thèses (avec Éric Laporte) dont une déjà soutenue :

- Elsa Tolone (encadrement à 50%, 2007-2011), actuellement en post-doc à l'Université de Cordoba en Argentine : intégration d'un lexique syntaxique riche dans un analyseur syntaxique symbolique
- Anthony Sigogne (encadrement à 80%, 2009-2012 ?) : intégration de lexiques syntaxiques dans des analyseurs syntaxiques probabilistes
- Myriam Rakho (encadrement à 80%, 2009-2013 ?) : annotation sémantique et multilingue de verbes

Les thèses d'Elsa Tolone et Myriam Rakho ne sont pas forcément au coeur de mes thématiques principales de recherche, mais sont des apports importants pour le projet collectif de l'équipe d'informatique linguistique du LIGM à laquelle j'appartiens. La thèse d'Anthony Sigogne est vraiment au coeur de mes problématiques actuelles et a fortement contribué à leur évolution.

J'ai publié dans deux communautés distinctes : la communauté de Traitement Automatique des Langues et celle de la Linguistique, avec des habitudes de publications différentes. Mes publications (avec sélection par un comité scientifique) comptent 3 revues, 2 articles de collections, 2 chapîtres de livre, 12 articles dans des actes de conférences internationales, 7 articles dans des actes d'ateliers internationaux et 9 articles d'actes de conférences nationales. J'ai également édité 2 actes de conférences et 1 collection d'articles. L'année 2012 a été particulièrement faste. J'ai été accepté à la conférence internationale ACL qui est la conférence la plus prestigieuse du Traitement Automatique des Langues. J'ai également obtenu le prix du meilleur article de la conférence TALN, la principale conférence francophone dans le domaine.

Chapitre 1

Introduction

1.1 Contexte et problématiques scientifiques

L'explosion du nombre de documents textuels disponibles, notamment via Internet, a rendu indispensable le développement d'applications permettant d'accéder le plus efficacement possible à des informations précises. En particulier, les moteurs de traduction ou les outils d'extraction d'informations (ex. informations biographiques, financières, opinions, mots clés, etc.) peuvent se montrer utiles. Ces applications sont de plus en plus souvent couplées à des analyseurs linguistiques, comme des étiqueteurs morphosyntaxiques ou des analyseurs syntaxiques, dont les bonnes performances les rendent désormais exploitables dans la vraie vie. Cependant, ces analyseurs ne tiennent pas, peu ou pas assez compte d'un type d'expressions qui foisonnent dans les textes : les expressions multi-mots.

Les expressions multi-mots (EMM), dans le consensus actuel du domaine du Traitement Automatique des Langues (TAL), forment des unités linguistiques complexes qui contiennent un certain degré de non-compositionalité lexicale, syntaxique, sémantique et/ou pragmatique. Elles regroupent les expressions figées, les collocations, les entités nommées, les verbes à particule, les constructions à verbe support, les termes, etc. Leur identification est donc fondamentale avant toute analyse sémantique. Par ailleurs, elle permet de réduire fortement la complexité combinatoire des textes [Gross et Senellart, 1998]. Ainsi, elle peut aider à améliorer les performances des analyseurs linguistiques. De même, leur prise en compte dans les applications est cruciale. Par exemple, la traduction de l'expression figée *boire les paroles de quelqu'un* en anglais est *to lap up with what somebody say.* Un système de traduction automatique ne doit donc pas chercher à traduire cette expression comme une séquence compositionnelle de mots : **drink the words of somebody*[1]. Par ailleurs, si un extracteur automatique de mots-clés sélectionne le mot simple *trou* dans un document d'Astronomie parlant d'un *trou noir*, on considérera cela comme une erreur car *trou noir* est un

1. Traduction trouvée par *Google Translate*.

terme qui doit être considéré dans son ensemble.

L'importance de telles expressions est bien connue de la communauté internationale du TAL depuis des décennies, par ex. [Gross, 1986,Heid, 1994]. Cependant, à part les entités nommées [2], elles ont longtemps été négligées. On a néanmoins vu émerger des travaux fondateurs sur l'aquisition automatique d'expressions multi-mots, comme [Smadja, 1993] et [Daille, 1995]. Il y aussi eu quelques tentatives isolées de constructions de ressources, par exemple dans le cadre du lexique-grammaire [Gross, 1994, D'Agostino et Vietri, 2004, Ranchhod et al., 2004] ou de la théorie sens-texte [Mel'cuk et al., 1999]. Des études pionnières ont intégré certains types d'expressions dans des analyseurs comme dans [Roche, 1997] et [Brun, 1998]. Depuis plus d'une dizaine d'années, on observe un regain d'intérêt de la communauté internationale, comme en témoigne l'organisation annuelle de l'atelier sur les expressions multi-mots, parrainé par l'Association for Computational Linguistics (ACL). Ce renouveau vient essentiellement de la prise de conscience de la communauté anglo-saxonne avec [Sag et al., 2002]. Bien qu'une très bonne part des publications concerne toujours le développement de techniques de repérage des entités nommées et d'acquisition automatique d'expressions multi-mots, les recherches se sont diversifiées : la constitution de ressources librement disponibles comme des lexiques [Kaalep et Muischnek, 2008, Grégoire, 2008] ou des corpus annotés [Laporte et al., 2008, Vincze et al., 2011b] ; l'intégration des expressions multi-mots dans différents formalismes grammaticaux [Copestake et al., 2002, Debusmann, 2004] ; des travaux sur leur interprétation sémantique [Kim et Baldwin, 2007, Nakov, 2008] ; leur traduction , etc. Malgré cela, la prise en compte des expressions multi-mots reste encore marginale dans les analyseurs empiriques. Des études ont, d'abord, montré qu'elle permettait d'améliorer les performances des analyseurs tout en supposant que les EMMs avaient été parfaitement reconnues au préalable [Nivre et Nilsson, 2004, Arun et Keller, 2005]. Enfin, récemment, des travaux ont cherché à incorporer une reconnaissance réaliste de sous-ensembles d'expressions multi-mots, e.g. [Cafferkey et al., 2007, Finkel et Manning, 2009, Korkontzelos et Manandhar, 2010, Wehrli et al., 2010, Green et al., 2011]. Cependant, nous n'en sommes qu'aux balbutiements.

Ainsi, notre premier axe de recherche a consisté à améliorer la finesse linguistique de l'annotation des analyseurs par la prise en compte des expressions multi-mots. Nos travaux ont porté à la fois sur les étiqueteurs morphosyntaxiques et sur les analyseurs syntaxiques. Par ailleurs, ses dernières années, nous avons assisté à l'essor sans précédent des méthodes probabilistes qui permettent d'obtenir les meilleurs outils actuels. L'intégration des expressions multi-mots dans ces outils revient essentiellement à adapter les modèles. Comme ces expressions sont, par définition, difficilement prédictibles, l'exploitation de ressources lexicales est primordiale pour leur reconnaissance. Nous avons donc été amené à trouver des stratégies d'intégration de ressources symboliques externes dans des modèles probabilistes associés à nos tâches. C'est ce que nous allons traiter dans

2. Les entités nommés ont commencé à être sérieusement étudiées dans le cadre de l'extraction d'information [Grishman et Sundheim, 1996].

les chapitres 2 et 3, respectivement dédiés à l'analyse de surface et à l'analyse syntaxique profonde. Nous y verrons aussi des techniques d'intégration de lexiques syntaxiques, pas forcément liées à la reconnaissance des expressions multi-mots. Naturellement, comme nous l'avons mentionné ci-dessus, les applications doivent aussi prendre en compte ces expressions. Durant nos premières années post-thèse, nous avons travaillé dans cette optique. Nous avons développé des applications liées au monde privé (extraction d'informations, classification) ou liées au monde académique (aide à la construction de lexiques bilingues ou à des études linguistiques). Dans tous les cas, nous nous sommes basé sur des prétraitements fins alimentés par des ressources lexicales riches. Le chapitre 4 est consacré à la présentation de nos différentes applications. Pour finir, le développement de tels analyseurs linguistiques ou applications n'est possible que grâce à l'existence de ressources (corpus annoté ou lexiques). Or, les ressources autour des multi-mots manquent ou sont incomplètes. Durant notre thèse principalement, nous nous sommes consacré à la construction de ressources contenant des expressions peu décrites (à l'époque) et liées aux entités nommées : expressions de mesures et groupes prépositionnels géographiques. Récemment, nous avons travaillé sur l'annotation de corpus. Le chapître 5 survole rapidement les différentes études descriptives que nous avons réalisées pour la construction de telles ressources. Il présente également l'outillage que nous avons mis en place pour gérer nos ressources lexicales.

1.2 Les expressions multi-mots

1.2.1 Définition et classification

Les expressions multi-mots, dans le consensus actuel du domaine du Traitement Automatique des Langues, forment des unités linguistiques qui contiennent un certain degré de non-compositionalité lexicale, syntaxique, sémantique et/ou pragmatique. Elles regroupent les expressions figées, les collocations, les entités nommées, les verbes à particule, les constructions à verbe support, les termes, etc. Elles apparaissent à différents niveaux de l'analyse linguistique : certaines forment des unités lexicales contigues à part entière (ex. *cordon bleu*, *San Francisco*, *tout à fait*, *par rapport à*), d'autres forment des constituants syntaxiques comme les phrases figées (*N0 prend le taureau par les cornes* ; *N0 cueillir N1 à froid* ; *N0 boire les paroles de N1*) ou les constructions à verbe support (*N0 donner un avertissement à N1* ; *N0 faire du bruit*).

Les classifications et terminologies de ces expressions sont très nombreuses et variées dans la littérature linguistique, par exemple [Fiala et al., 1997, Cowie, 1998, Grossmann et Tutin, 2003, Burger et al., 2007, Mel'cuk, 2011]. Dans ce mémoire, nous ne rentrons pas dans la discussion. A titre d'exemple, nous noterons que la classification la plus populaire actuellement dans la communauté internationale du TAL est celle décrite dans [Sag et al., 2002]. Ces derniers proposent de découper les expressions multi-mots en deux classes : les expressions lexicalisées (*lexicalized phrases*) et les expressions institutionalisées (*institutionalized*

phrases). Les expressions lexicalisées possèdent un certain degré de figement syntaxique et/ou sémantique, qui peut être détecté par des critères linguistiques formels. Les expressions institutionalisées sont compositionnelles syntaxiquement et sémantiquement, mais sont statistiquement idiosyncratiques : les mots des expressions apparaissent ensemble soit par convention soit de manière habituelle (ex. *traffic jam*).

Nous présentons maintenant quelques genres d'expressions multi-mots, qui nous seront utiles dans la suite : les expressions figées, les constructions à verbe support, les collocations et les entités nommées. Pour de plus amples descriptions, nous renvoyons à la littérature sur le sujet.

Expressions figées Les expressions figées sont des combinaisons de plusieurs mots, non-compositionnelles du point de vue sémantique : ex. *cul de sac*, *s'envoyer en l'air*. Les critères linguistiques pour déterminer si une combinaison de mots est une expression figée sont basés sur des tests syntaxiques et sémantiques comme ceux décrits dans [Gross, 1982, Gross, 1986]. Par exemple, l'expression *boîte noire* est une expression figée car elle n'accepte pas de variations lexicales (**boîte sombre*, **caisse noire*) et elle n'autorise pas d'insertions (**boîte très noire*). De même, l'expression *casser sa pipe* avec le sens de 'mourir' n'accepte pas de variations lexicales comme **(rompre+briser) sa pipe* ou **casser son fume-cigarette*. Elle n'autorise pas de transformations comme la passivation **Sa pipe a été cassée par Max*.

Tout d'abord, il existe des expressions figées qui forment des séquences contigues de mots. On traite ces expressions comme des unités lexicales, de manière équivalente à des mots simples. On peut imaginer que l'on remplace les espaces par des tirets : *cordon bleu* → *cordon-bleu*. Elles peuvent prendre des structures hétérogènes (*poule mouillée*, *pied à terre*, *nid de poule*, un *je ne sais quoi*). On les appelle des mots composés. Ils forment des unités lexicales auxquelles on peut associer une partie-du-discours : ex. *tout à fait* est un adverbe, *à cause de* est une préposition, *table ronde* est un nom. Les variations morphologiques et lexicales sont très limitées – e.g. *caisse noire+caisses noires*, *vin (rouge+blanc+rosé+*orange+...)* – et les variations syntaxiques très souvent interdites [3] (**caisse très noire*). Leur opacité sémantique est variable : par exemple, *cordon bleu* est totalement opaque ; *vin rouge* est plus transparent [4]. Le degré de figement peut être détecté au moyen de batteries de tests syntaxiques comme dans [Gross, 1996a].

Il existe également des constructions verbales figées qui ne peuvent être considérées comme des unités lexicales à part entière et qui forment des constituants syntaxiques. Prenons, par exemple, une séquence figée entièrement lexicalisée comme *les carottes sont cuites* qui, au premier abord, pourrait s'analyser comme un seul mot. Cependant, on observe certaines variations syntaxiques comme des insertions adverbiales :

3. De telles expressions acceptent parfois des insertions, souvent limitées à des modifieurs simples e.g. *à l'insu de*, *à l'insu justement de*.

4. Un *vin rouge* est un vin, mais n'est pas vraiment de couleur rouge. L'ensemble dénote un type de vin.

(1) *Les carottes, avec Max, sont toujours cuites.*

(2) *En janvier prochain, les carrottes seront cette fois entièrement cuites.*

Tout comme les phrases libres, certaines phrases figées acceptent des arguments libres. Ceux-ci peuvent même parfois s'insérer à l'intérieur de la séquence figée.

(3) *N0 casser les oreilles à N1 = : Max casse les oreilles à Marie.*

(4) *N0 cueille N1 à froid = : Max cueille Marie à froid.*

De même, certaines expressions acceptent des transformations, comme la passivation, d'autres pas :

(5) *Marie a été cueillie à froid par Max.*

(6) *Luc porte Lea dans son coeur = * Lea est portée dans son coeur par Luc.*

La limite des phrases figées n'est pas clairement définie. Par exemple, certaines expressions sont figées syntaxiquement mais acceptent une forte variation lexicale d'un des arguments [Gross, 1988]. Prenons les expressions suivantes :

(7) *N0 faire DU (ski+natation+piano+...)*

Dans ces expressions, la structure syntaxique est figée et, en particulier, le déterminant. Les noms forment une classe sémantiquement homogène des noms d'activités. Ce type d'expression se situe à la limite des phrases figées car elles sont compositionnelles sémantiquement, mais figées syntaxiquement.

Constructions à verbe support Les constructions à verbe support mettent généralement en jeu un nom prédicatif et un verbe appelé support [Gross, 1981]. Le verbe support a un sens vide ou réduit mais peut parfois être remplacé par un verbe qui porte une valeur sémantique (ex. valeur aspectuelle). Le nom prédicatif est soit en position objet, soit en position sujet [Danlos, 2009]. Le nom garde son sens habituel. C'est lui qui porte le sens de la construction et qui sélectionne les arguments. Dans le premier exemple ci-dessous, le verbe support est *faire*, le nom prédicatif est *confiance*.

(8) *Marie fait confiance à Léa*

(9) *Luc (a+ressent) de l'affection pour Léa*

(10) *Max fait du bruit*

Ces constructions ont des propriétés syntaxiques particulières. Tout d'abord, le nom garde son autonomie et la construction peut se réduire en un groupe nominal où le verbe support disparaît :

(11) *L'affection de Luc pour Léa = L'affection que Luc ressent pour Léa*

Ensuite, une double analyse est possible pour les compléments, alors que ce n'est pas le cas avec un verbe plein. Dans les exemples ci-dessous, *mener* est un verbe support et *raconter* est un verbe plein.

(12) *Le général (mène+raconte) une attaque contre le fort.*

(13) *C'est (une attaque contre le fort) que le général (mène+raconte).*

(14) *C'est (une attaque) que le général (mène+*raconte) contre le fort.*

On notera également des contraintes sur le déterminant dépendant du nom.

(15) *Marie fait (E^5+*une+*sa) confiance à Léa*

(16) *Le général mène (*E+une+*son) attaque contre le fort*

Ce type de constructions a été largement étudié dans le cadre du lexique-grammaire : ex. [Giry-Schneider, 1978, Giry-Schneider, 1987, Gross, 1989]. On notera également les études dans le cadre de la théorie sens-texte, où ces constructions sont décrites à l'aide de fonctions lexicales [Mel'cuk, 1998]. Cette notion a également été intégrée dans le cadre de la construction du lexique FrameNet [Fillmore et al., 2003].

Collocations Le terme *collocation* est utilisé à toutes les sauces dans la littérature du TAL. Mais à quoi correspond-il exactement ? Les collocations sont décrites comme des combinaisons de mots qui présentent des affinités et tendent à apparaître ensemble (pas forcément de manière contigue) [Tutin et Grossmann, 2002] : par exemple, *argument de poids*, *amour fou*, *les prix s'envolent*. Le sens de ces expressions est relativement transparent, i.e. il se devine assez facilement. Par contre, elles sont difficiles à produire pour un non-natif. Il existe deux approches principales pour définir les collocations. Tout d'abord, en linguistique de corpus, les collocations sont considérées comme des combinaisons habituelles de mots au sens fréquentiel [Sinclair, 1991]. Cette définition est celle utilisée le plus souvent par les chercheurs en TAL qui spécifient les collocations à l'aide de mesures associatives statistiques [Smadja, 1993, Pecina, 2010]. Elle est assez large et couvre toutes les expressions multi-mots. Il existe une autre approche plus statisfaisante d'un point de vue lexicographique. Selon [Tutin et Grossmann, 2002], "une collocation est une cooccurrence lexicale privilégiée de deux éléments linguistiques entretenant une relation syntaxique". Comme le soulignent [Haussman, 1979, Mel'cuk, 2011], l'un des constituants (la base) garde son sens habituel et le sens de l'autre (le collocatif) est déterminé lexicalement par la base. Par contre, ils entretiennent des relations syntaxiques compositionnelles (*Luc a battu un record = un record a été battu par Luc = c'est un record que Luc a battu*). Une collocation est donc souvent qualifiée de semi-compositionnelle. Elle se situe entre les expressions libres et les expressions figées.

La frontière avec les expressions figées n'est pas toujours très claire. En particulier, certaines collocations sont relativement opaques comme, par exemple, *peur bleue*. Avec les critères utilisés dans le cadre du lexique-grammaire, ce type d'expressions serait considéré comme un mot composé : *peur (bleue+*rouge+*orange)*, **peur très bleue*, **sa peur est bleue*, etc. Dans la suite, nous considérerons ce type de collocations nominales comme des mots composés. Il en est de même pour

5. Le symbole E indique le mot vide.

la collocation *battre un record* qui se trouve dans les tables des figés du lexique-grammaire [Gross, 1982]. Dans certains cadres [Tutin et Grossmann, 2002], les collocations intègrent également les constructions à verbe support. En effet, le nom prédicatif joue le rôle de la base qui a son sens habituel et le verbe support celui du collocatif qui, dans ce cas, se vide entièrement ou partiellement de son sens. Devant le flou autour de la notion de collocation, nous éviterons le plus souvent d'utiliser ce terme. En cas d'obligation, nous spécifierons sa définition : par exemple, *collocation de type statistique* ou *collocation de type lexicographique*.

Entités nommées Les entités nommées sont des phénomènes qui ont été largement étudiés dans le TAL (cf. [Ehrmann, 2008]) car ce sont des unités fondamentales pour l'extraction d'information [Grishman et Sundheim, 1996]. Il existe diverses classifications comme celle de [Sekine et al., 2002] qui fait référence dans le monde du TAL. Les entités nommées comprennent de nombreux phénomènes linguistiques comme les noms propres (noms de personne, d'organisation, etc.), les expressions numériques ou les expressions de temps. Certaines entités doivent être répertoriées sous la forme de listes pour être reconnues et classifier (ex. *San Francisco*). D'autres peuvent être décrites à l'aide de grammaires spécifiques comme les dates (ex. *le 5 mars 2010*) ou les noms de personne (ex. *Jacques Chirac*). Etant donné cette grammaire, ces expressions sont sémantiquement compositionnelles et peuvent être normalisées.

1.2.2 Identification

L'identification des expressions multi-mots dans les textes est souvent complexe car elles sont difficilement prédictibles automatiquement. Dans cette partie, nous présentons les principales approches pour les identifier. La plupart du temps, leur reconnaissance est fondée sur la consultation de lexiques. Ces derniers peuvent être acquis soit manuellement, soit par des méthodes automatiques. Les méthodes automatiques sont généralement fondées sur des mesures statistiques qui servent à filtrer les candidats (ex. [Dunning, 1993, Dias, 2003, Pecina, 2010, Ramisch et al., 2010b]). Elles sont souvent combinées avec des traitements linguistiques comme l'étiquetage morphosyntaxique ou l'analyse syntaxique. Par exemple, certains utilisent des patrons syntaxiques basés sur un étiquetage grammatical pour restreindre les candidats [Daille, 1995, Watrin, 2006, Ramisch et al., 2010a]. D'autres, comme le suggérait [Heid, 1994], se fondent sur une analyse syntaxique afin de capturer aussi les variations syntaxiques de certaines expressions [Seretan et al., 2003] comme les collocations verbe-nom (au sens *lexicographique*). On observe l'émergence de classifieurs binaires [Ramisch et al., 2010a, Tu et Roth, 2011]. Etant donné un candidat, le classifieur indique si ce candidat est une expression multi-mot d'un certain type : les termes pour [Ramisch et al., 2010a], les constructions à verbe support pour [Tu et Roth, 2011]. Les modèles de classification incorporent différents types de traits. Par exemple, ils peuvent contenir des traits statistiques (ex. mesures associatives) ou des traits plus contextuels (les mots mis en relation, leurs étiquettes grammaticales, les valeurs des mots, des informations provenant

de ressources externes – ex. les classes de Levin [Levin, 1993] dans [Tu et Roth, 2011] –). Une autre méthode consiste à se servir de corpus alignés pour extraire des expressions multi-mots comme dans [Caseli et al., 2010, Zarrieß et Kuhn, 2009].

Le plus grand désavantage des stratégies entièrement basées sur des lexiques est que cette procédure est incapable de découvrir de nouvelles expressions. Certains comme [Vincze et al., 2011a] ont donc proposé de combiner, par disjonctions et conjonctions, plusieurs critères basés, entre autres, sur des lexiques, des patrons syntaxiques, des relations syntaxiques, etc. On assiste aussi à l'émergence d'approches probabilistes supervisées. Celles-ci obtiennent des résultats *état-de-l'art* comme [Green et al., 2011, Vincze et al., 2011b]. Par exemple, [Vincze et al., 2011b] utilisent les champs aléatoires markoviens (CRF) et [Green et al., 2011] utilisent une grammaire à substitution d'arbres. Ces méthodes ont l'avantage d'être capables d'apprendre de nouvelles expressions. Dans ce mémoire, nous montrerons, en particulier, l'intérêt de combiner l'approche probabiliste supervisée avec la consultation de ressources lexicales.

1.3 Analyse syntaxique

On distingue généralement deux types d'analyse syntaxique : (a) l'analyse de surface qui revient à une tâche d'annotation syntaxique en segments et (b) l'analyse profonde qui revient à une tâche d'annotation syntaxique sous forme de graphes (ex. arbres de constituants ou arbres de dépendances). Dans cette section, nous décrivons les principaux formalismes, modèles et méthodes utilisés pour ces deux tâches. Nos recherches décrites dans ce mémoire ont essentiellement consisté à adapter ces modèles et ces méthodes dans le but d'intégrer la reconnaissance des expressions multi-mots et d'exploiter différentes ressources lexicales.

1.3.1 Analyse de surface

L'analyse linguistique de surface se compose traditionnellement de l'étiquetage morphosyntaxique suivie de l'analyse en constituants simples (ou analyse en *chunks*). L'étiquetage morphosyntaxique affecte une étiquette à chaque mot d'une phrase, alors que l'analyse en chunks découpe la phrase en constituants syntaxiques simples. Ces deux tâches ont été très largement étudiées dans la littérature. Avec l'augmentation des puissances de calcul, la conception de modèles probabilistes de plus en plus fins et la mise à disposition de corpus annotés de référence, les approches statistiques ont, depuis quelques années, connu un essor sans précédent dans le domaine et ont permis de développer les outils les plus performants. L'étiquetage morphosyntaxique et l'analyse en constituants simples utilisent de plus en plus des modèles discriminants multi-traits comme les champs markoviens aléatoires (CRF) [Lafferty et al., 2001, Sha et Pereira, 2003], les modèles maximum d'entropie (MaxEnt) [Ratnaparkhi, 1996, Toutanova et al., 2003] ou les séparateurs à vaste marge (SVM) [Kudo et

Matsumoto, 2001, Gimenez et Marquez, 2004]. Par exemple, pour l'anglais, les étiqueteurs et les analyseurs en chunks obtiennent des performances de l'ordre de 97% et 94% respectivement sur le Wall Street Journal Penn Treebank [Marcus et al., 1993]. Certaines études récentes, comme [Denis et Sagot, 2009] pour l'étiquetage morphosyntaxique, ont montré que l'intégration de ressources lexicales externes dans de tels outils améliorait significativement leurs performances. Bien que moins performants en qualité d'annotation, les « vieux » modèles génératifs de Markov caché (HMM) et les dérivés des modèles n-grammes restent néanmoins populaires de part leur simplicité d'implantation et leur flexibilité [Schmid, 1994, Ramshaw et Marcus, 1995, Thede et Harper, 1999, Brants, 2000]. Avec l'essor des machines à états finis dans le TAL durant les années 90 [Mohri, 1997, Kornai, 1999, Karttunen, 2001], de nombreux travaux ont cherché à implanter avec succès ces modèles à l'aide de cette technologie [Mohri, 1997], notamment [Nasr et Volanschi, 2005] pour l'analyse en constituants simples. Par ailleurs, des analyseurs superficiels sous la forme de cascades ou compositions de transducteurs ont vue le jour : par exemple, [Abney, 1996, Federici et al., 1996, Ait-Mokhtar et Chanod, 1997]. En plus de leur grande flexibilité, cette technologie permet une intégration simple de ressources lexicales [Silberztein, 1994, Nasr et al., 2010]. Les machines à états finis ont permis de développer des étiqueteurs et analyseurs à base de règles comme dans [Roche et Schabes, 1995, Karlsson et al., 1995, Laporte et Monceaux, 1999].

1.3.2 Analyse profonde

L'analyse syntaxique consiste à structurer un texte sous la forme d'un graphe qui établit les relations syntaxiques entre les mots ou des groupes de mots. Dans nos travaux, nous nous sommes intéressés à deux types : (a) les analyseurs en constituants qui produisent une structure en constituants hiérarchisés sous la forme d'arbre ; (b) les analyseurs en dépendance qui relient directement les mots entre eux lorsqu'ils dépendent syntaxiquement l'un de l'autre. Durant nos travaux, nous nous sommes particulièrement intéressés aux analyseurs probabilistes. Tout d'abord, les meilleurs analyseurs en constituants sont basés sur le formalisme des grammaires algébriques probabilistes (PCFG). Leurs performances ont été particulièrement améliorées depuis une quinzaine d'années avec le développement de différentes stratégies. Tout d'abord, certains ont proposé de lexicaliser les grammaires en annotant les symboles non-terminaux par leur tête lexicale, e.g. [Charniak, 1997, Collins, 2003]. D'autres ont proposé d'atténuer le problème de l'indépendance des hypothèses entre les règles, en introduisant au niveau des symboles non-terminaux soit des informations provenant du contexte comme dans [Johnson, 1998, Klein et Manning, 2003], soit des symboles latents (cachés) calculées automatiquement comme dans [Matsuzaki et al., 2005, Petrov et al., 2006]. Les grammaires avec annotations latentes, appelées PCFG-LA, obtiennent actuellement des résultats *état-de-l'art*, et, en particulier, pour le français [Seddah et al., 2009, Le Roux et al., 2011]. Enfin, de récentes études ont montré que coupler plusieurs grammaires chacune apprise avec un biais différent sur le même corpus améliorait très significativement les performances [Petrov,

2010]. Pour diminuer le problème de la dispersion lexicale, des expériences ont montré l'intérêt de remplacer les mots par des classes plus générales afin de réduire la complexité lexicale. La classe associée à un mot peut être son lemme, une forme désinfléchie [6] [Candito et Crabbé, 2009], une classe provenant d'un réseau sémantique ou d'un thesaurus [Agirre et al., 2008], une classe calculée automatiquement avec un algorithme de classification non supervisée appliqué sur un corpus brut [Candito et Crabbé, 2009], etc. Un tel analyseur syntaxique peut être couplé avec un réordonnanceur qui prend en entrée les n meilleures analyses d'un analyseur lambda et les reclasse en fonction de traits non-locaux comme dans [Charniak et Johnson, 2005] à l'aide d'un modèle MaxEnt ou dans [Huang, 2008] avec un perceptron. Cette méthode permet encore d'augmenter la qualité d'analyse de manière très significative. Par ailleurs, au milieu des années 2000, les analyseurs en dépendance ont connu un grand bond en avant, avec le développement de méthodes basées sur des modèles discriminants comme [Yamada et Matsumoto, 2003, Nivre, 2003, Mcdonald et al., 2005]. Récemment, l'exploitation des affinités lexicales comme dans [Bansal et Klein, 2011, Mirroshandel et al., 2012] a montré des résultats très prometteurs.

En dehors des analyseurs probabilistes, il existe des analyseurs symboliques dont les grammaires sont construites à la main. Ils sont basés sur des formalismes grammaticaux riches : par exemple, les grammaires syntagmatiques guidées par les têtes (HPSG) [Pollard et Sag, 1994], les grammaires lexicales-fonctionnelles (LFG) [Kaplan et Bresnan, 1982], les grammaires d'arbres adjoints (TAG) [Joshi et al., 1975], les grammaires en dépendance par ex. [Gerdes et Kahane, 2001, Duchier et Debusmann, 2001]. Un analyseur symbolique peut être couplé à un lexique morphologique et syntaxique à large couverture. Par exemple, pour le français, les analyseurs FRMG [Thomasset et de La Clergerie, 2005] et SxLFG [Boullier et Sagot, 2005] se basent sur le lexique Lefff [Sagot, 2010]. De nombreux travaux ont été réalisés sur le sujet, notamment le développement de formalismes et l'étude de l'interaction entre la grammaire et le lexique. Pour faciliter la construction manuelle de la grammaire, un système de métagrammaire peut permettre d'éliminer certaines redondances : ex. XMG [Crabbé, 2005]. Les formalismes mis en oeuvre doivent être capables de tenir compte des expressions multi-mots : e.g. HPSG [Copestake et al., 2002], TAG [Abeillé, 1993, Schuler et Joshi, 2011], LFG [Danlos et al., 2006], grammaires en dépendance [Debusmann, 2004]. Les analyseurs symboliques produisent toutes les analyses possibles. Il faut donc leur ajouter un module de désambiguation, soit à base d'heuristiques comme dans FRMG [Thomasset et de La Clergerie, 2005] soit à base d'estimateurs probabilistes comme dans [Riezler et al., 2002]. Durant sa thèse que j'ai co-encadrée, E. Tolone a en particulier été amenée à travailler avec FRMG.

6. Une forme désinfléchie est une forme à laquelle on enlève les marques morphologiques tout en préservant son ambiguité grammaticale.

1.3.3 Ressources

Dans cette section, nous présentons succintement les différentes ressources que nous avons exploitées durant nos recherches. Nous nous limitons strictement aux ressources de notre langue de travail, le français, qui est l'une des langues les mieux dotées en terme de ressources morphologiques et syntaxiques. Nous avons utilisé trois types de ressources lexicales : (a) des dictionnaires électroniques ; (b) des grammaires locales fortement lexicalisées ; (c) des lexiques syntaxiques. Les dictionnaires et les grammaires locales ont été essentiellement utiles pour l'étiquetage morphosyntaxique et la reconnaissance d'expressions multi-mots. Les lexiques syntaxiques ont servi pour des expériences cherchant à intégrer des ressources lexicales dans des analyseurs syntaxiques en profondeur.

Les dictionnaires électroniques sont des listes d'entrées lexicales. Chaque entrée contient sa forme fléchie, son lemme, sa catégorie grammaticale, des informations morphologiques (ex. genre et nombre), des informations sémantiques (ex. trait humain pour les noms). Ces dictionnaires comprennent non seulement des mots simples mais aussi des mots composés. Ils sont compressés sous la forme de transducteurs finis pour être appliqués de manière efficace aux textes [Silberztein, 1994]. Les principaux dictionnaires que nous avons utilisés sont librement disponibles : les dictionnaires de langue générale DELA [Courtois, 2009, Courtois et al., 1997] et Lefff [Sagot, 2010], un dictionnaire de toponymes Prolex [Piton et al., 1999], etc. L'ensemble de ces dictionnaires comportent au total au dessus d'un million d'entrées. Les mots composés sont majoritairement codés dans le DELA avec plus de 200 000 entrées. Les entrées composées y sont aussi associées à leur structure interne. Par exemple, *eau de vie* a le code `NDN` car sa structure interne est *nom – de – nom*.

Les grammaires locales [Gross, 1997] sont fondées sur le formalisme des réseaux de transitions récursives (RTN) [Woods, 1970] et reconnaissent théoriquement des langages algébriques. Elles sont essentiellement utilisées pour décrire et reconnaitre des expressions multi-mots contigues. En particulier, il est possible de représenter des classes syntaxiques comme les déterminants nominaux ou des classes syntaxico-sémantique telles que les adverbes de temps. Elles se fondent sur l'application préalable de dictionnaires électroniques et se représentent sous la forme de graphes d'automates finis [Silberztein, 1994] sur un alphabet composé de symboles terminaux et non-terminaux. Un symbole terminal est soit un mot soit un masque lexical. Un masque lexical correspond à une entrée lexicale sous-spécifiée (i.e. des traits sont manquants). Par exemple, le masque <*noun+plural*> correspond à toutes les entrées lexicales (des dictionnaires) qui sont des noms au pluriel. Enfin, les symboles non-terminaux sont des références à d'autres graphes. En pratique, les grammaires locales fortement lexicalisées reconnaissent des langages rationnels et peuvent donc être compilées en automates finis. Nous avons à notre disposition des centaines de graphes qui reconnaissent par exemple des entités nommées [Martineau et al., 2009], des déterminants numériques et nominaux [Silberztein, 2003, Laporte, 2007], des prépositions locatives, etc.

Un lexique syntaxique décrit pour chaque prédicat son comportement syn-

taxique propre. Il en existe de nombreux librement disponibles pour le français. Historiquement, les tables du lexique-grammaire [Gross, 1975] correspondent au premier lexique syntaxique du français. C'est d'ailleurs le plus riche en terme d'entrées lexicales et de propriétés syntaxiques codées. Il se représente sous une forme tabulaire. Malheureusement, ce lexique n'a pas été conçu directement pour le TAL et ne peut pas être exploité tel quel. Nous avons donc utilisé d'autres lexiques disponibles, dont deux directement dérivés des tables du lexique-grammaire :

- DicoValence [Van den Eynde et Mertens, 2003] : il a été construit manuellement au moyen de l'approche pronominale
- LGLex [Constant et Tolone, 2010a] : il est issu directement des tables du lexique-grammaire [Gross, 1975, Gross, 1994]
- Lefff [Sagot, 2010] : il a été construit semi-automatiquement à partir de diverses sources
- LGLex-Lefff [Tolone, 2011] : il correspond au lexique LGLex qui a été converti au format Lefff au moyen d'heuristiques
- LexSchem [Messiant et al., 2008] : il a été construit automatiquement à partir d'un corpus brut

Tous les lexiques (sauf LGLex) ont en commun qu'ils intégrent, pour chaque entrée lexicale, un cadre de sous-catégorisation avec les arguments liés à leurs fonctions syntaxiques (ex. sujet, obj, objde, etc.) et leur production syntagmatique (ex. groupe nominal, complétive, infinitive, etc.). LGLex associe, à chaque entrée lexicale, un ensemble de propriétés (ex. nature des arguments, distribution lexicale des prépositions, constructions syntaxiques, transformations, etc.). Tous les lexiques décrivent à la base les verbes du français. Le Lefff contient également des informations syntaxiques sur les adjectifs. LGLex et LGLex-Lefff décrivent aussi le comportement de noms qui entrent dans des constructions à verbe support. LGLex contient, en plus, plus de 30 000 constructions verbales figées que nous n'avons pas encore exploitées dans nos travaux.

Pour apprendre et évaluer les différents modèles de nos outils, nous nous sommes servi du corpus arboré de Paris 7 [7] [Abeillé et al., 2003] qui est annoté en constituants syntaxiques. Il est composé d'articles provenant du journal *Le Monde*. Les mots composés y sont marqués et forment au total plus de 5% des unités lexicales (mots simples et composés). Il existe plusieurs versions de ce corpus. Nous en avons utilisé trois : (a) la version "originale" comprenant près de 19 000 phrases ; (b) la version FTB corrigée de juin 2010 [8] qui comprend près de 16 000 phrases ; (c) la version corigée FTB-UC [Candito et Crabbé, 2009] dont les noms composés à la structure syntaxique régulière (ex. N A ou N P N) ont été déliés, ce qui a réduit de moitié le nombre de mots composés. Il existe également un corpus annoté en dépendance FTB-DEP [Candito et al., 2010] converti directement du FTB-UC.

7. http ://www.llf.cnrs.fr/Gens/Abeille/French-Treebank-fr.php
8. Cette version nous a été envoyée par Marie Candito (Univ. Paris 7, Alpage).

Chapitre 2

Analyse de surface

L'analyse linguistique de surface se compose traditionnellement de l'étiquetage morphosyntaxique suivi de l'analyse en constituants syntaxiques simples (ou analyse en *chunks*). Nous avons vu, dans l'introduction, que les meilleurs étiqueteurs et chunkers basés sur des modèles discriminants obtenaient désormais de très bons résultats. Ces résultats peuvent même être significativement améliorés si l'on exploite des ressources lexicales, notamment pour aider au traitement des mots inconnus [1] dans l'étiquetage morphosyntaxique [Denis et Sagot, 2009]. Tous ces étiqueteurs et analyseurs ont néanmoins un défaut : ils ne tiennent pas ou peu compte des expressions multi-mots [2]. Tenir compte de ces expressions revient à considérer ces deux tâches comme des tâches (jointes) de segmentation et d'étiquetage. Cela ne change pas pour l'analyse en *chunks*. Par contre, c'est nouveau pour l'étiquetage morphosyntaxique. Par exemple, la phrase *Jean boit de l'eau de vie* serait segmentée et étiquetée *Jean/NPP boit/V de_l'/DET eau_de_vie/NC*. Or, il est bien connu que le caractère imprévisible des expressions polylexicales les rend difficilement repérables sans l'aide de ressources lexicales. Dans ce chapître, nous proposons un cadre général d'intégration de ressources externes dans un segmenteur-étiqueteur. Nous appliquons ensuite ce cadre aux deux tâches liées à l'analyse de surface, l'étiquetage morphosyntaxique et l'analyse en constituants simples. Nous nous limitons à la reconnaissance d'expressions multi-mots de niveau lexical, équivalentes à des unités lexicales. Nous avons donc naturellement exploité des dictionnaires morphosyntaxiques de mots simples et composés ainsi que des grammaires locales fortement lexicalisées.

1. Les mots inconnus sont les mots absents du corpus d'apprentissage.

2. On notera cependant l'expérience dans [Korkontzelos et Manandhar, 2010] qui montrent que la pré-identification de quelques types d'expressions multi-mots permettait d'améliorer les performances d'un analyseur en constituants simples.

2.1 Intégration de ressources lexicales

2.1.1 Stratégies de base

La littérature fait apparaître deux stratégies principales pour intégrer des ressources lexicales externes dans des systèmes d'annotation : (a) limitation de l'*espace de recherche*, i.e. limitation à l'espace des analyses trouvées dans les ressources ; (b) utilisation de *traits exogènes* dans les modèles discriminants, i.e. des traits liés aux ressources. Nous montrons maintenant les approches préliminaires que nous avons utilisées pour implanter ces deux stratégies.

Limitation de l'espace de recherche

La stratégie de la limitation de l'espace de recherche est très naturellement implantable dans une infrastructure à états finis. En collaboration avec O. Blanc et P. Watrin, nous avons mis au point une architecture classique basée sur des cascades de transducteurs finis [Blanc et al., 2007]. Elle est composée de trois phases distinctes : (1) analyse ambigue par cascade de transducteurs qui génère un automate (nommé TFST) représentant les différentes analyses possibles ; (2) élagage partiel de l'automate en utilisant différents modules de levée d'ambiguité ; (3) linéarisation de l'automate par application de l'algorithme du plus court chemin [3] se basant sur une pondération de l'automate elle-même fondée sur un modèle. Cette architecture est synthétisée dans la figure 2.1.

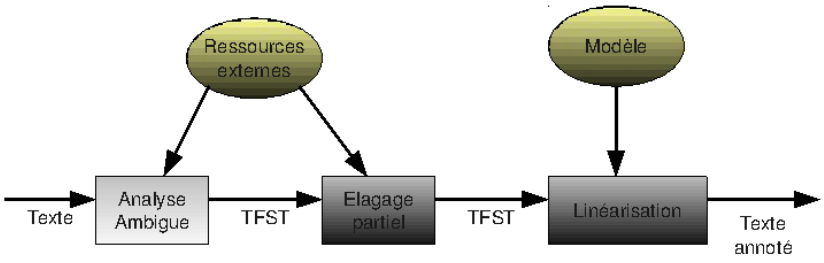


FIGURE 2.1 – Architecture à états-finis

La phase (1) comporte obligatoirement une analyse lexicale suivie optionnellement d'une analyse en constituants simples dans le cas d'un chunker. Elle est entièrement fondée sur des ressources lexicales et grammaticales externes. Elle est implantée sous la forme d'une cascade de transducteurs et construit de manière itérative un automate représentant l'ensemble des analyses trouvées ou reconnues par nos ressources. A chaque fois qu'un transducteur reconnait

3. Le plus court chemin est le chemin qui a le plus petit poids. Le poids d'un chemin est le produit des poids de ses transitions.

une séquence, il ajoute l'analyse correspondante dans l'automate. Les analyses ajoutées à une étape peuvent être utilisées aux étapes suivantes. L'analyse lexicale a la particularité qu'elle autorise l'analyse des expressions multi-mots continues (présentes dans nos dictionnaires ou reconnues par nos grammaires locales). La figure 2.2 donne un exemple de résultat d'une telle analyse. Nous reprenons ainsi l'approche proposée dans les plateformes Intex [Silberztein, 2000], Unitex [Paumier, 2003] et NooJ [Silberztein, 2008]. Cette stratégie d'analyse lexicale est similaire à celle du système Macaon [Nasr et al., 2010]. Cependant, ce dernier ne permet pas l'intégration de grammaires locales.

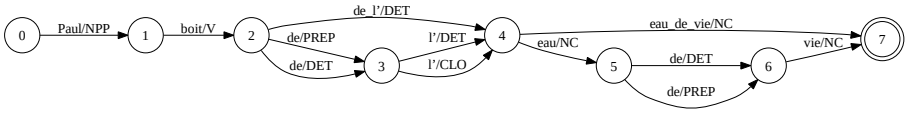


FIGURE 2.2 – Analyse lexicale

La phase (2) d'élagage permet d'affiner encore plus l'espace de recherche en se basant sur des modules d'élagage fins ou des heuristiques sur lesquels l'utilisateur a un certain contrôle : par exemple, un élagueur à base de règles (externes). Ces modules peuvent la plupart du temps s'implanter avec des algorithmes classiques des automates finis [Blanc et al., 2007, Sigogne, 2010]. La phase (3) de linéarisation permet de sélectionner le meilleur chemin en fonction d'un "modèle" fourni par l'utilisateur. Ce modèle permet de pondérer l'automate auquel on applique un algorithme du plus court chemin. Dans nos premières expériences [Blanc et al., 2007], nous avons développé un modèle "maison" dans lesquels les poids étaient soit fixés manuellement selon l'étiquette soit fixés au moyen de statistiques basiques. Ce modèle "maison" s'est ensuite mué en modèle probabiliste : HMM [Sigogne, 2010] et CRF [Constant et Sigogne, 2011]. Théoriquement, pour certains types de modèles, la phase de pondération peut se réaliser par composition de transducteurs : e.g. pour HMM [Nasr et Volanschi, 2005] et pour CRF [Constant et Sigogne, 2011]. En pratique, on applique l'algorithme de Viterbi qui est plus efficace et plus simple à mettre en oeuvre.

Le système hybride proposé permet d'avoir la main lors de toutes les étapes de l'analyse. Nous évitons ainsi l'effet boîte noire des analyseurs purement statistiques. Nous évitons également une trop grande dépendance vis-à-vis d'un corpus annoté de référence en cas de linéarisation stochastique. Cette infrastructure a d'abord été mise en place pour de l'analyse en constituants avec reconnaissance des expressions multi-mots pour l'écrit [Blanc et al., 2007], puis pour l'oral [Blanc et al., 2010]. Elle a été étendue à l'étiquetage morphosyntaxique [Sigogne, 2010, Constant et Sigogne, 2011].

Utilisation de traits exogènes

Pour l'incorporation de *traits exogènes*, nous nous sommes inspirés des travaux de [Denis et Sagot, 2009] sur l'étiquetage morphosyntaxique. En collaboration

avec A. Sigogne et I. Tellier, nous avons mis en place des stratégies similaires, cette fois appliquées à des segmenteur-étiqueteurs, dans lequels il faut tenir compte de la segmentation contrairement à l'étiquetage pur. Pour cela, nous nous sommes encore basés sur une analyse lexicale poussée. L'automate généré TFST ne sert pas cette fois à limiter l'espace de recherche, mais plutôt à extraire des traits pour des modèles probabilistes discriminants. Lors de nos expériences préliminaires [Constant et al., 2011, Constant et Tellier, 2012], nous nous sommes ramenés à une annotation équivalente de schéma *BIO* [Ramshaw et Marcus, 1995]. Chaque mot est étiqueté par X-TAG où TAG est l'étiquette de l'unité lexicale à laquelle appartient le mot et X précise la position relative du mot dans l'unité : B au début, I dans les autres positions. Cela signifie que l'automate TFST doit être converti dans ce schéma. L'automate de la figure 2.2 est transformé en l'automate déterminisé de la figure 2.3.

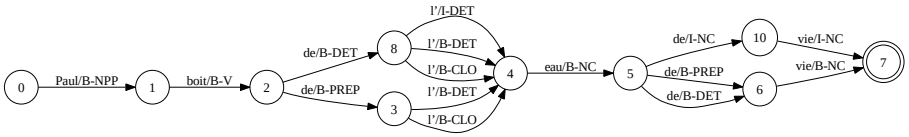


FIGURE 2.3 – Analyse lexicale dans le schéma BIO

Chaque position dans la séquence peut être associée à un ensemble de transitions, que l'on peut ramener à un ensemble d'étiquettes. Par exemple, pour la position correspondant aux transitions sortant de l'état 5 (le mot *de*), on a l'ensemble d'étiquettes {I-NC,B-DET,B-PREP}. Il y a deux moyens de se servir de ces informations dans les traits. Soit on se base sur un attribut correspondant à la concaténation de ces étiquettes dans un certain ordre (ex. ordre alphabétique), ce qui revient à une classe d'ambiguité : *B-DET/B-PREP/I-NC* [stratégie *learn-concat*]. Soit on se base sur des attributs booléens, chacun d'entre eux correspondant à une étiquette du jeu d'étiquettes [stratégie *learn-bool*]. Si l'étiquette est posssible à la position donnée, alors l'attribut est vrai, sinon il est faux. Les attributs générés pour notre exemple (cf. figure 2.3) sont donnés dans la table 2.1 pour la stratégie *learn-concat* et dans la table 2.2 pour la stratégie *learn-bool*. Théoriquement, le jeu d'étiquettes de l'automate ne doit pas forcément correspondre à celui de l'espace de recherche. Cependant, en pratique, on utilise le même jeu. [Denis et Sagot, 2009] ont d'ailleurs montré que c'était la meilleure configuration pour leur tâche d'étiquetage morhosyntaxique.

Il est également possible de se baser sur l'automate TFST qui a été partiellement ou totalement élagué (cf. phases 2 et 3 de la sous-section *limitation de l'espace de recherche*). Par exemple, en appliquant un algorithme qui garde les plus courts chemins en s'aidant d'une pondération homogène (i.e. chaque transition a le même poids), on obtient un automate totalement ou quasiment linéarisé. Cette analyse correspond à une (ou plusieurs) segmentation(s) naïve(s) où les analyses multi-mots sont favorisées. La procédure d'extraction de traits se baserait alors sur l'automate TFST linéarisé de la figure 2.4.

mot	concaténation
Paul	B-NPP
boit	B-V
de	B-DET/B-PREP
l'	B-CLO/B-DET/I-DET
eau	B-NC
de	B-DET/B-PREP/I-NC
vie	B-NC/I-NC

TABLE 2.1 – Texte avec les attributs de la stratégie *learn-concat*

mot	B-CLO	B-DET	B-NC	B-NPP	B-PREP	B-V	I-DET	I-NC
Paul	0	0	0	1	0	0	0	0
boit	0	0	0	0	0	1	0	0
de	0	1	0	0	1	0	0	0
l'	1	1	0	0	0	0	1	0
eau	0	0	1	0	0	0	0	0
de	0	1	0	0	1	0	0	1
vie	0	0	1	0	0	0	0	1

TABLE 2.2 – Texte avec les attributs de la stratégie *learn-bool*

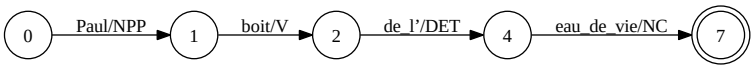


FIGURE 2.4 – Automate TFST linéarisé par l'algorithme des plus courts chemins

Les stratégies que nous avons expérimentées sous-exploitent la structure de TFST. En particulier, on extrait des attributs qu'en fonction de la position de la séquence à segmenter-étiqueter. On pourrait avoir envie de calculer des attributs en fonction d'une transition dans l'automate. Par exemple, à partir de l'automate de la figure 2.3, on peut vouloir extraire les étiquettes possibles à la position précédente de celle de la transition *vie/I-NC* en tenant compte des contraintes de TFST. Dans ce cas, il n'y en a qu'une possible *I-NC*, issue de la transition *de/I-NC*. Dans la prochaine section, nous allons présenter un cadre général qui autorise naturellement de formuler ce genre de traits.

2.1.2 Un cadre général

En collaboration avec I. Tellier, nous avons donc formulé un cadre général d'intégration de ressources externes [Constant et Tellier, subm] en se basant sur nos études empiriques précédentes (cf. section 2.1.1). Notre proposition est une reformulation du problème de l'annotation séquentielle dans un cadre formalisé.

Principe

Pour assurer une certaine généricité, la représentation des données (exemples étiquetés et ressources externes) ne doit pas être liée à un quelconque outil d'apprentissage. Il est aussi nécessaire de trouver un langage permettant de combiner les informations provenant de différentes origines. Nous nous sommes naturellement tournés vers les bases de données relationnelles. Celles-ci permettent de stocker tous types de données. Elles sont indépendantes de toute application. Enfin, le langage SQL offre un moyen puissant d'extraire, combiner ou mettre à jour les informations stockées dans les bases de données relationnelles.

L'annotation séquentielle consiste à trouver une séquence $y = y_1 y_2 ... y_n$ d'étiquettes à une séquence $x = x_1 x_2 ... x_n$ d'éléments. Par exemple, l'étiquetage morphosyntaxique consiste à affecter une catégorie grammaticale à chaque mot d'une séquence. Dans notre cas, on suppose que l'on dispose d'un certain nombre de données hétéroclites : exemples étiquetés, lexiques, sorties de prétraitement (ex. automate TFST issu de l'analyse lexicale, etc.). Toutes ces données peuvent être représentées dans des bases de données. Dans [Constant et Tellier, subm], nous montrons que les requêtes SQL permettent de nous ramener à une base de données relationnelle composée de trois tables. Celles-ci permettent aussi d'accéder aux informations présentes dans cette unique base de données. En particulier, elles servent à extraire les traits des modèles probabilistes, réécrire les données dans un format exploitable par un logiciel, etc. Les trois tables sont les suivantes :

ITEM(item-id,value,item-att$_1$, ..., item-att$_n$)
ITEM-POSITION(pos-item-id,seq-id,pos,item-id,pos-item-att$_1$,...,pos-item-att$_m$)
LABEL-POSITION(pos-label-id,seq-id,pos,label,pos-label-att$_1$,....,pos-label-att$_l$)

Dans la table ITEM, chaque enregistrement correspond aux caractéristiques propres d'un élément (ex. mot), indépendamment de ses différentes positions dans les séquences. Par exemple, les attributs propres des mots sont les suffixes et les préfixes de certaines tailles, la valeur booléenne indiquant si le mot commence par une majuscule, si le mot se trouve dans une liste, etc. Dans la table, l'indentifiant d'un élément *item-id* joue le rôle de clé primaire, et sa valeur lexicale est dans le champs *value*. Les caractéristiques propres correspondent aux attributs *item-att*$_1$, ..., *item-att*$_n$. On donne un exemple d'ITEM dans la table 2.3.

item-id	word	lowercase	1-suffix	2-suffix	1-prefix	isCapitalized
item1	Paul	paul	l	ul	P	1
item2	boit	boit	t	it	b	0
item3	de	de	e	de	d	0
item4	l'	l'	'	l'	l	0
item5	eau	eau	u	au	e	0
item6	vie	vie	e	ie	v	0

TABLE 2.3 – Table ITEM

Dans la table ITEM-POSITION, chaque enregistrement correspond à un

élément plongé dans un contexte séquentiel. Chaque élément *item-id* est associé à sa position *pos* dans une séquence *seq-id*. Elle possède également des attributs *pos-item-att*$_1$,..., *pos-item-att*$_m$ qui codent des informations dépendant de l'élément et de sa position. Par exemple, la table peut correspondre aux résultats générés par une analyse lexicale. Dans ce cas précis, chaque enregistrement représenterait une transition de l'automate TFST. Il est aussi possible de ne se limiter qu'à une partie des informations présentes dans TFST : par exemple, la table 2.4 contient un attribut directement issu de TFST et qui correspond à la stratégie *learn-concat* (cf. section 2.1.1). L'attribut BOS indique si l'élément est au début de la phrase.

pos	item-id	BOS	learn-concat
1	item1	1	B-NPP
2	item2	0	B-V
3	item3	0	B-DET/B-PREP
4	item4	0	B-CLO/B-DET/I-DET
5	item5	0	B-NC
6	item3	0	B-DET/B-PREP/I-NC
7	item6	0	B-NC/I-NC

TABLE 2.4 – Table POSITION-ITEM

Dans la table LABEL-POSITION, chaque enregistrement correspond à une étiquette de sortie *label* associé à une position *pos* dans une séquence $seq - id$, avec certaines contraintes liées à l'espace de recherche (*pos-label-att*$_1$,...., *pos-label-att*$_l$). A la phase d'apprentissage supervisé, l'espace de recherche est résolu : il n'y a qu'une seule étiquette par position. En cas d'apprentissage non supervisé ou semi-supervisé, cet espace n'est pas forcément résolu. A la phase d'étiquetage, l'espace de recherche peut être l'ensemble des analyses possibles dans le jeu d'étiquettes. Il peut aussi avoir été partiellement résolu (limitation par analyse lexicale, pré-étiquetage partiel, élagage partiel après analyse lexicale). Par exemple, l'espace de recherche peut être représenté par un automate. Dans ce cas-là, un enregistrement correspond à une transition. Par exemple, la table 2.5 illustre l'espace de recherche spécifié par l'analyse lexicale comme dans l'automate de la figure 2.3. En particulier, les attributs *src-state* et *dest-state* sont les identifiants des états source et destination des transitions.

pos-label-id	pos	label	src-state	dest-state
poslabel1	1	B-NPP	0	1
poslabel2	2	B-V	1	2
poslabel3	3	B-DET	2	8
poslabel4	3	B-PREP	2	3
poslabel5	4	I-DET	8	4
poslabel6	4	B-DET	8	4
poslabel7	4	B-CLO	8	4
poslabel8	4	B-DET	3	4
poslabel9	4	B-CLO	3	4
poslabel10	5	B-NC	4	5
...	...	...	...	...

TABLE 2.5 – Table POSITION-LABEL

Ainsi, un tel cadre permet d'intégrer exemples étiquetés et ressources lexicales de manière simple et formalisée. Elles peuvent être combinées dans différents buts, notamment : limiter l'espace de recherche (dans la table POSITION-LABEL) ; utiliser l'ensemble des tables pour produire les caractéristiques du modèle probabiliste au moyen de requêtes SQL.

Application aux CRF linéaires

Nous montrons maintenant comment appliquer notre cadre général aux champs aléatoires markoviens linéaires. Dans de tels modèles, un trait est défini par une fonction caractéristique binaire $f(t, x, y_t, y_{t-1})$ où t est la position courante dans x. Par exemple, le trait f_{1222} est défini par la fonction :

$$f_{1222}(t, x, y_t, y_{t-1}) = \begin{cases} 1 & \text{if } x_{t-2} = \text{"boit" and } y_t = \text{"I-DET"} \\ 0 & \text{else} \end{cases}$$

Cette fonction peut être reformulée avec une requête SQL qui sera appliquée à notre base de données DB. On note DB_t une vue de DB qui exprime, en fonction de t, les contraintes liées aux CRF linéaires sur x et y pour définir une fonction caractéristique. Cette vue inclut les tables ITEM et POSITION-ITEM car tout le contenu de x est disponible quel que soit la position t. Par contre, elle contient une vue de POSITION-LABEL qui incorpore uniquement les informations sur y en positions t et $t-1$. La fonction caractéristique met en jeu un vecteur v et une requête r qui est appliquée sur la vue DB_t à la position t ($pos = t$). Si le vecteur v est inclus dans le résultat de l'application de la requête r, le trait est activé, c'est-à-dire la fonction associée retourne 1. Une fonction caractéristique peut donc être reformulée de la manière suivant :

$$f(t, DB_t) = \begin{cases} 1 & \text{if } q(t, DB_t) \text{ contains } r \\ 0 & \text{else} \end{cases}$$

Notre exemple f_{1222} devient alors :

$$f_{1222}(t, x, DB_t) = \begin{cases} 1 & \text{if } query33(t, DB_t) \text{ contains } \begin{array}{|c|c|}\hline \textit{word} & \textit{label} \\ \hline \text{boit} & \text{I-DET} \\ \hline \end{array} \\ 0 & \text{else} \end{cases}$$

La requête $query33$ serait implantée de la manière suivante, avec $t qui indique la position t :

```
SELECT word, label
FROM POSITION–ITEM, ITEM, POSITION–LABEL
WHERE POSITION–LABEL.position=$t AND POSITION–ITEM.position=
    POSITION–LABEL.position - 2 AND POSITION–ITEM.item-id = ITEM.
    item-id
```

Les fonctions caractéristiques sont créées lors de la phase d'entrainement à partir de la base de données correspondant au corpus d'entrainement. A chaque

position de chaque séquence d'apprentissage, on applique chaque requête fournie par l'utilisateur. Chacune des applications des requêtes génère un tableau (une liste de vecteurs). Pour chacun de ces vecteurs, nous créons une fonction caractéristique qui met en jeu la requête et le vecteur.

2.2 Etiquetage morphosyntaxique

Cette section est dédiée aux expériences que nous avons réalisées sur l'étiquetage morphosyntaxique. Elles ont conduit au développement de deux segmenteurs-étiqueteurs [4] hydrides librement disponibles sous licence LGPL : *HybridTagger*[5] [Sigogne, 2010] et *lgtagger* [Constant et Sigogne, 2011]. *HybridTagger* utilise un modèle probabiliste HMM et *lgtagger* un CRF linéaire. Ces deux outils sont fondés sur une analyse lexicale par consultation de dictionnaires de mots simples et composés et, optionnellement, par application de grammaires locales fortement lexicalisées, comme dans la section précédente. L'automate TFST issu de l'analyse lexicale sert à la limitation de l'espace de recherche et à l'extraction de traits pour le modèle CRF linéaire.

2.2.1 Un étiqueteur basé sur les modèles de Markov cachés

L'étiqueteur *HybridTagger*, développé par A. Sigogne, est fondé sur l'architecture à états finis décrite dans la section 2.1.1 et qui peut être naturellement intégrée dans le cadre général de la section 2.1.2. Nous avons donc repris les trois phases : (1) analyse ambigue ; (2) élagage partiel ; (3) linéarisation. Tout d'abord, la phase d'analyse ambigue est limitée à l'analyse lexicale. Afin de rendre l'étiqueteur robuste, on considère que les mots simples absents dans nos ressources peuvent avoir toutes les étiquettes possibles dans le jeu d'étiquettes. L'automate TFST issu de l'analyse lexicale est ensuite élagué partiellement. Nous avons utilisé deux modules : l'un statistique LearningErrors [Sigogne, 2010] et l'autre symbolique ELAG [Laporte et Monceaux, 1999]. Le module statistique est dérivé de l'étiqueteur de Brill [Brill, 1995]. Ce dernier permet de corriger un étiquetage initial au moyen de règles de transformations instanciées à partir du corpus d'apprentissage et d'un ensemble de patrons de règles lexicales ou contextuelles. Dans notre cas, nous utilisons les règles, non pas pour corriger, mais pour élaguer l'automate de la phrase. A chaque position, on essaye d'appliquer chaque règle. Si celle-ci est applicable, on supprime toutes les hypothèses d'étiquettes, sauf celle précisée dans la règle. Comme dans [Brill, 1995], l'ordre d'application des règles est important. Celui-ci est calculé à la phase d'apprentissage. Au total, 950 règles lexicales et 19 contextuelles ont été générées. Le module symbolique est ELAG [Laporte et Monceaux, 1999] qui applique un ensemble de règles écrites manuellement sur un automate ambigü. Nous avons pris en compte la totalité

4. Un segmenteur-étiqueteur est un étiqueteur morphosyntaxique intégrant la reconnaissance des expressions multi-mots.

5. *HybridTagger* a été développé par A. Sigogne lors de son stage de Master 2 que j'encadrais. L'étiqueteur a été en partie intégré à la plate-forme *Unitex* [Paumier, 2003].

des 45 règles définies dans les grammaires ELAG originelles ainsi que 6 autres créées au cours du développement de l'étiqueteur. Enfin, l'automate TFST est linéarisé à l'aide des modèles de Markov cachés (HMM). Notons que les probabilités liées aux mots inconnus sont calculées en fonction de leurs suffixes de manière similaire à [Brants, 2000]. HybridTagger a été évalué sur le FTB-UC en supposant que tous les mots composés étaient parfaitement reconnus. On constate une amélioration très significative de l'étiquetage des mots inconnus par rapport à d'autres étiqueteurs du même type comme TnT [Brants, 2000] ou Treetagger [Schmid, 1995]. A. Sigogne a aussi montré que chacun des modules d'élagage améliorait la précision de l'étiqueteur.

2.2.2 Un étiqueteur basé sur les champs markoviens aléatoires

Le segmenteur-étiqueteur *lgtagger*[6] a initialement été développé dans l'infrastructure à états-finis de la section 2.1.1, cf. [Constant et Sigogne, 2011]. La phase d'analyse ambigue est limitée à l'analyse lexicale. Afin de rendre l'outil robuste, on considère que les mots simples inconnus dans nos ressources peuvent avoir toutes les étiquettes possibles dans le jeu d'étiquettes. Nous utilisons aussi des grammaires chargées de deviner des mots composés non présents dans nos ressources purement lexicales. Par exemple, un nom propre peut être une séquence de mots commençant par une majuscule. La phase d'élagage est ignorée. Enfin, la phase de linéarisation est fondée sur un modèle CRF linéaire. L'automate TFST issu de la phase d'analyse lexicale y est transformé en un automate TFST-BIO équivalent dans le schéma *BIO* [Ramshaw et Marcus, 1995]. TFST-BIO est alors pondéré au moyen de compositions de transducteurs, en s'inspirant de [Nasr et Volanschi, 2005]. Un algorithme du plus court chemin est ensuite appliqué pour trouver la meilleure analyse. Le modèle CRF inclut, non seulement des traits classiques de l'étiquetage morphosyntaxique, mais aussi des *traits exogènes*. Les traits exogènes sont calculés selon la stratégie *learn-concat* (cf. section 2.1.1). Les évaluations sur le corpus FTB ont montré des résultats *état-de-l'art* pour *lgtagger*. Nous avons montré que la prise en compte des expressions multi-mots ne pouvait pas être négligée. On constate une chute de plus de 3 points par rapport à l'étiquetage supposant une reconnaissance parfaite des expressions multi-mots : 97.7% vs. 94.4%. Par ailleurs, l'intégration de ressources lexicales améliore significativement les résultats de l'ordre de +0.7 point. Quant à l'identification des mots composés (avec traits exogènes), elle atteint des performances de l'ordre de 75-78% en terme de F-mesure selon le jeu d'étiquettes utilisé. Si l'on n'utilise pas de traits exogènes, on constate une chute des performances de plus de 4 points. Par contre, on constate que l'application du modèle avec limitation de l'*espace de recherche* a des performances légèrement moindres que dans un mode sans limitation. Ceci s'explique par le fait qu'avec limitation de l'espace, l'outil n'est pas capable de reconnaître des

6. Il est fondé sur les programmes d'Unitex [Paumier, 2003] pour l'application de ressources lexicales et sur le logiciel Wapiti [Lavergne et al., 2010] pour l'apprentissage et l'application des modèles CRF.

unités multi-mots inconnues ne se trouvant pas dans nos ressources lexicales, ce qui n'est pas le cas dans un mode sans limitation.

Par la suite, le mode avec limitation de l'espace de recherche a donc été abandonnée [Constant et Tellier, 2012]. La phase de linéarisation par des compositions de transducteurs et l'algorithme du plus court chemin a aussi été abandonné au profit d'un algorithme de Viterbi classique plus efficace (en temps de calcul). L'intégration de ressources lexicales ne s'est donc limitée qu'à l'incorporation de *traits exogènes*. Parmi les trois tables de notre base de données relationnelles, nous n'agissons donc plus que sur les tables ITEM et POSITION-ITEM de notre cadre général. Par ailleurs, dans [Constant et Tellier, 2012], nous avons montré que la stratégie d'intégration *learn-concat* obtenait des résultats légèrement meilleurs que *learn-bool*. De plus, le modèle correspondant à cette dernière approche est plus gourmand en espace mémoire et il est plus lent à apprendre. La combinaison des deux stratégies ne donne rien de significatif. Dans [Constant et al., 2011], nous avons également testé la stratégie qui consiste à ajouter les entrées de nos ressources comme de nouveaux exemples. Ceci revient à rajouter des enregistrements à toutes les tables lors de la phase d'apprentissage. Cependant, sans surprise, cette méthode donne de moins bons résultats que les autres stratégies car de multiples biais statistiques sont introduits avec cette méthode.

Ainsi, la meilleure stratégie d'intégration de ressources lexicales est *learn-concat* sans limitation de l'espace de recherche. Dans [Constant et Tellier, 2012, Constant et Tellier, subm], nous observons que l'intégration des ressources lexicales permet d'améliorer l'étiquetage des mots simples et composés inconnus de manière très significative (+3 points dans [Constant et Tellier, subm]). Dans [Constant et Sigogne, 2011, Constant et Tellier, subm], nous montrons aussi que notre système obtient des résultats *états-de-l'art* en le comparant à d'autres outils tels que Treetagger (3.7 points), SVMTool (1.6 points) et MeLT (2 points), qui ne sont pas à l'origine conçus pour intégrer la reconnaissance des unités polylexicales.

Enfin, nous avons montré que l'intégration de ressources lexicales externes permet de compenser la petite taille d'un corpus d'apprentissage. En effet, dans la figure 2.5, on observe que notre système entrainé sur 40% du corpus d'apprentissage du FTB en intégrant nos ressources lexicales a des performances équivalentes à ce même système entrainé sur tout le corpus d'entrainement sans intégrer les ressources lexicales.

2.3 Analyse en constituants simples

2.3.1 Le super-chunker POM

L'article initial de notre approche à états-finis [Blanc et al., 2007] décrit un analyseur en constituants simples (ou *chunks*), nommé POM, qui intègre la reconnaissance des expressions multi-mots continues (ex. mots composés, entités nommées, etc.). Ainsi, nous y étendons la notion de *chunk* classique dans le sens de [Abney, 1991] à la notion de *super-chunk* qui autorise la présence

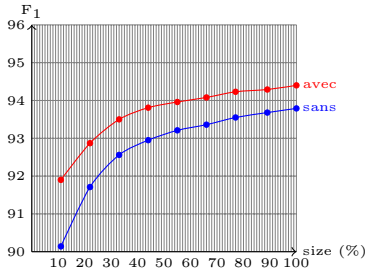


FIGURE 2.5 – Evolution des performances d'un segmenteur-étiqueteur en fonction de la taille du corpus d'apprentissage. La courbe avec (resp. sans) correspond à l'expérience avec intégration des ressources lexicales (resp. sans).

d'unités polylexicales. Par exemple, la séquence *la marge d'exploitation* est annotée comme un super-chunk nominal (XN), alors que, dans l'analyse classique, elle serait annotée par une séquence de deux constituants simples : un groupe nominal (*la marge/XN*) suivi d'un groupe prépositionnel (*d'exploitation/XP*). L'exemple 17 illustre un découpage en *super-chunks*. La phrase contient quatre unités polylexicales (entre parenthèses) : l'adverbe de temps *durant le premier trimestre 2007*, le déterminant nominal *l'ensemble des*, le nom *chiffre d'affaires brut* et le déterminant numérique *6 121 millions de*. Elle est composée de 6 super-chunks au lieu de 11 chunks dans l'analyse classique.

(17) [(Durant le premier trimestre 2007)], [(l'ensemble des) activités] [au Maroc] [ont généré] [un (chiffre d'affaires brut)] [de (6 121 millions de) dirhams]

Ainsi, plusieurs unités polylexicales peuvent être combinées dans un même super-chunk car n'importe quelle unité lexicale peut être une expression multimots. C'est le cas dans la phase suivante :

[XN La température] [XP (à l'intérieur de) (beaucoup de) maisons] [XP en Moldavie]

Le constituant *à l'intérieur de beaucoup de maisons* est considéré comme un super-chunk prépositionnel (XP) car *à l'intérieur de* est une préposition composée, *beaucoup de* est un déterminant composé et *maisons* est un nom simple.

Les super-chunks verbaux sont également spécifiques car ils incluent les verbes modaux, les insertions, les clitiques, la négation ainsi que les auxiliaires au sens de [Gross, 1999a]. Par exemple, on a l'annotation suivante pour la phrase *Jean n'a pas pu les trouver* :

[XN Jean] [XV n'a pas pu les trouver]

La séquence discontinue *n'... pas* est une négation, *a ... pu* est la forme au passé composé du verbe modal *pouvoir* et *les* est un clitique accusatif.

L'analyseur que nous avons développé est fondé sur l'architecture en trois phases décrite dans la section 2.1.1 : (1) analyse ambigue ; (2) élagage partiel ; (3) linéarisation. La phase (1) d'analyse ambigue est composée de l'analyse lexicale puis de l'analyse en chunks. On réalise l'analyse lexicale par consultation de dictionnaires de mots simples et composés puis par l'application en cascade de grammaires locales fortement lexicalisées. Cette phase génère, pour chaque phrase, un automate représentant l'ensemble des analyses compatibles avec nos ressources lexicales que ce soit pour les mots simples ou les unités polylexicales. Puis, nous réalisons une analyse en chunks ambigue. Comme dans [Abney, 1996], nous appliquons une cascade de grammaires locales reconnaissant itérativement les différents consituants syntaxiques simples. A chaque fois qu'un chunk est reconnu dans l'automate, il est rajouté dans celui-ci avec une nouvelle transition. Ces grammaires implémentent un mécanisme d'héritage des traits des unités au niveau des constituants : par exemple, la tête d'un consituant nominal est la valeur lexicale de son dernier nom (simple ou composé) ; le temps d'un constituant verbal est celui de sa tête verbale.

La phase (2) d'élagage partiel contient deux modules symboliques : l'un basé sur une heuristique simple, l'autre basé sur des règles écrites manuellement. Le module fondé sur l'heuristique, appelé *SPH*, cherche à favoriser les analyses avec unités polylexicales (i.e. les unités les plus longues). Pour cela, il applique l'algorithme des plus courts chemins sur l'automate de la phrase où toutes les transitions ont le même poids (pondération uniforme). On ne garde donc dans l'automate que les chemins les plus courts (de même longueur). Le module basé sur les règles manuelles est appelé *Luberon*. Il applique itérativement une liste de règles qui fonctionnent de la manière suivante. Etant donné une ambiguité et, optionnellement, un contexte droit ou/et gauche, la règle résoud l'ambiguité en sélectionnant la première étiquette dans la liste. En pratique, à chaque état de l'automate, on vérifie si l'ambiguité de la règle correspond à la liste des transitions sortantes de l'état. Si le contexte droit ou/et gauche correspond aussi, on ne garde que la transition correspondant au premier élément de l'ambiguité. Ainsi, l'ordre de l'application des règles est très important. C'est parfois un difficile dosage à réaliser. Par exemple, si on a l'ambiguité XN-XP avec XP introduit par *de* et comme contexte gauche direct un XN, on voudra sélectionner préférentiellement un XP. Donc on éliminera la transition correspondant à l'analyse XN.

Enfin, la phase (3) de linéarisation se fonde sur l'algorithme du plus court chemin : on garde uniquement le plus court chemin. Dans le cas où il existe plusieurs plus courts chemins de même poids, on en choisit un au hasard. Le poids de chaque transition est calculé très simplement. Il correspond à la probabilité d'associer la tête du constituant correspondant à la transition, avec sa catégorie de chunk. Dans le cas où la valeur lexicale de la tête est inconnue, on donne un poids dépendant uniquement de la catégorie grammaticale. Dans son mémoire de stage de Master 2 [Sigogne, 2009], A. Sigogne a modifié la pondération du module de linéarisation en se basant sur les modèles de Markov caché

(HMM). Les probabilités d'émissions correspondent à la probabilité d'une étiquette étant donné le mot introducteur et la tête du constituant. Les probabilités de transitions correspondent aux trigrammes d'étiquettes de chunks.

Dans [Blanc et al., 2007], nous montrons d'excellents résultats sur un corpus de dépêches AFP : environ 95% de F-mesure. Les erreurs sont équitablement réparties entre la couverture des ressources lexicales, les modules d'élagage *SPH* et *Luberon*, la linéarisation. Ces excellents scores s'expliquent par le grand effort réalisé pour ajuster les ressources lexicales et grammaticales, ainsi que les règles de Luberon, à ce type de corpus (assez stéréotypé). Dans [Sigogne, 2009], A. Sigogne a évalué POM sur un corpus bien plus difficile à analyser, nommé MIX : un mélange du roman "Le Tour du Monde en 80 jours" de Jules Verne et des rapports du parlement français. Ce corpus a été annoté semi-automatiquement par S. Voyatzi et T. Nakamura. Il constate que les performances chutent spectaculairement à 76% avec l'analyseur POM originel. Ceci s'explique en partie par le manque de couverture lexicale, notamment au niveau des termes du parlement et des entités nommées souvent complexes dans le roman de Jules Vergne. Par exemple, il observe que, si l'on rajoute dans nos lexiques les termes multi-mots du langage parlementaire, on augmente les performances de 2 points, quelle que soit la pondération du module de linéarisation. L'utilisation de la pondération HMM permet de gagner plus de 2 points (78.2%). A. Sigogne a également montré que le module d'élagage le plus efficace (et de loin) était le module SPH basé sur une heuristique assez naïve (mais performante).

Par ailleurs, en collaboration avec A. Dister, nous avons adapté notre analyseur aux transcriptions orales [Blanc et al., 2010]. Ces transcriptions sont remplies de phénomènes spécifiques qui viennent perturber l'analyse automatique. Ces phénomènes sont généralement appelés des disfluences : les répétitions (*le le chien*), les autocorrections immédiates (*la le chien*), les amorces corrigées ou non corrigés, etc. [Dister et al., 2009a]. [Benzitoun et al., 2004] estiment que l'annotation de corpus oraux n'est pas un problème spécifique étant donné qu'il n'existe pas de grammaire pour le langage parlé (vs. une grammaire pour le langage écrit) [Blanche-Benveniste et al., 1990]. Ainsi, ceci tend à montrer que si l'on supprime les disfluences, il n'est pas forcément nécessaire de modifier les grammaires ou règles grammaticales d'un analyseur existant (conçu pour l'écrit) pour annoter des textes oraux. C'est l'approche que nous avons adoptée. Nous avons donc mis au point un module de prétraitement des transcriptions avec repérage et nettoyage des disfluences (cf. section 4.2.1). L'analyseur POM a ainsi pu être utilisé tel quel. Les expériences sur un corpus provenant du corpus Valibel [Dister et al., 2009b] ont montré des performances de l'ordre de 84%. Ces scores sont comparables avec ceux obtenus sur des corpus de même type par les meilleurs analyseurs en constituants simples de la campagne EASY [Paroubek et al., 2007]. Nous avons aussi montré que l'enrichissement des ressources lexicales par une trentaine de mots spécifiques à l'oral permettait de gagner 2 points. Notons que notre approche se distingue de celle décrite dans [Antoine et al., 2008]. Ces derniers ont mis au point un analyseur en constituants simples se fondant sur une cascade de transducteurs en deux étapes sans pré-repérage des disfluences : une première étape avec une grammaire de chunk classique ne

TABLE 2.6 – Exemple d'analyse en super-chunks

WORD	POS	CHUNK
l'	B-DET	B-XN
ensemble	I-DET	I-XN
des	I-DET	I-XN
activités	B-N	I-XN
ont	B-V	B-XV
généré	B-V	I-XV
121	B-DET	B-XN
millions	I-DET	I-XN
de	I-DET	I-XN
dirhams	B-N	I-XN

tenant pas compte des spécificités de l'oral, puis une deuxième étape corrigeant la première phase en tenant compte de ces spécificités.

Le système que nous avons développé permet donc d'intégrer naturellement des ressources lexicales adaptées à nos besoins. L'infrastructure proposée permet d'intégrer des approches statistiques assez répandues telles que HMM au moyen de l'algorithme du plus court chemin. Cependant, nous observons un point négatif dans notre architecture. Lors de la pondération, il n'est pas vraiment possible de tenir compte de la séquence des transitions (i.e. des séquences d'étiquettes grammaticales) qui ont été traversées pour identifier un chunk. Dans notre système, nous ne pouvons faire remonter que des informations ciblées (ex. têtes des constituants, prépositions introductives, etc.). Notons que le système par composition de transducteurs proposé par [Nasr et Volanschi, 2005] permet de tenir compte de la pondération de l'étiquetage grammatical pour pondérer une analyse en chunks en utilisant des compositions de transducteurs. Dans la section suivante, nous allons essayer de compenser en partie ce problème en combinant l'analyseur POM avec un analyseur basé sur un modèle CRF.

2.3.2 Combinaison avec les champs markoviens aléatoires

Plus récemment, dans [Constant, subm], nous avons appliqué notre cadre général à l'analyse en constituants simples incluant la reconnaissance des unités polylexicales, soit l'analyse en *super-chunks*. Notre objectif était d'améliorer un analyseur basé sur les champs markoviens aléatoires, en se servant de la sortie générée par l'analyseur POM décrit dans la section précédente.

L'analyse en chunks et, par extension, l'analyse en super-chunks revient à une tâche d'annotation séquentielle [Ramshaw et Marcus, 1995]. Chaque mot est associé à une étiquette X-TAG où TAG est l'étiquette du super-chunk auquel le mot appartient et X correspond à la position relative du mot dans le super-

chunk (B pour le début et I pour les autres positions). La table 2.6 contient un exemple de phrase annotée en super-chunks : la colonne CHUNK correspond à l'étiquette de sortie de l'analyseur.

Plusieurs études [Lafferty et al., 2001, Tsuruoka et al., 2009] ont montré que les CRF linéaires permettent d'obtenir des performances *état-de-l'art* pour l'analyse en *chunks*. Généralement, les modèles intègrent des traits calculés à partir des mots eux-mêmes ainsi que leurs parties-du-discours prédites par un étiqueteur, comme dans [Tsuruoka et al., 2009]. Cependant, les analyseurs classiques ne prennent pas en compte les unités multi-mots. Dans la section précédente, nous avons vu que l'identification de telles unités pouvait être réalisée avec succès conjointement à l'étiquetage grammatical. La colonne CAT de la table 2.6 correspond à une telle tâche. L'extension d'un chunker à un super-chunker par modèle CRF peut donc consister à utiliser de telles étiquettes plutôt qu'un étiquetage simple. Dans notre cadre de travail, cela revient à rajouter un champs CAT dans la table POSITION-ITEM. Pour le calcul des traits, nous utilisons les requêtes correspondantes des patrons décrits dans [Tsuruoka et al., 2009] pour leur chunker de base : unigrammes, bigrammes et trigrammes de mots et d'étiquettes grammaticales. Le segmenteur-étiqueteur utilisé pour remplir le champs CAT a été appris en utilisant les traits décrits dans [Constant et Tellier, 2012]. Ainsi, lorsque l'on évalue ce modèle sur le FTB, on obtient des performances de l'ordre de 90% en terme de F-mesure, ce qui est tout à fait honorable étant donné que l'étiquetage morphosyntaxique avec reconnaissance des unités polylexicales est de l'ordre de 94%.

Le problème des annotations du FTB est qu'elles ne correspondent pas exactement aux annotations attendues pour l'analyseur POM. Avec POM, on considère que toutes les unités multi-mots codées dans les ressources correspondent à nos besoins, même si, bien sûr, il y a du silence. Or, de nombreuses unités polylexicales de nos ressources ne sont pas marquées comme telles dans le FTB. Par ailleurs, les super-chunks verbaux de POM ne correspondent pas exactement aux noyaux verbaux du FTB car ils peuvent inclure des auxiliaires au sens de [Gross, 1999b]. Ainsi, si l'on applique notre super-chunker basé sur CRF sur le corpus d'évaluation MIX ont obtient un score de 76% en terme de F-mesure soit autant que POM, ce qui est prometteur. Par contre, en terme de segmentation pure, POM est bien meilleur (+4 points), c'est dans l'étiquetage que celui-ci pêche. Notre idée est donc de se servir de POM pour guider l'analyseur CRF en terme de segmentation en super-chunks. Nous avons utilisé deux méthodes. Avec la première (FUSION), on combine les sorties des deux analyseurs afin d'obtenir la meilleure annotation possible. Avec la deuxième méthode (CRF-adapté), on adapte d'abord le corpus d'apprentissage aux annotations attendues en combinant la sortie de POM avec l'annotation de référence. Puis, on ajoute des nouveaux champs dans la table POSITION-ITEM correspondant à l'étiquetage et la segmentation générée par POM. Ceci vont servir comme source de traits supplémentaires dans le modèle CRF. On apprend ensuite un nouveau modèle pour l'analyseur. La méthode de combinaison de deux annotations est simple. Pour la segmentation, on fusionne les deux annotations dans un même graphe. Chaque noeud correspond à un super-chunk à une position donnée. Un

	CRF-base	POM	FUSION	CRF-adapté
U_1	80.5	84.7	**87.9**	**88.2**
F_1	76.0	76.1	83.2	**83.6**

TABLE 2.7 – Performances de l'analyse en super-chunks.

arc lie deux super-chunks c1 et c2 si la fin de c1 coïncide avec le début de c2 dans la texte. La segmentation finale est alors le plus court chemin dans le graphe. En cas d'ambiguité, c'est la segmentation de POM qui est favorisée. Au niveau de l'étiquetage des super-chunks, c'est l'étiquette de référence ou de l'analyseur CRF qui est favorisée en cas d'ambiguité.

Nous synthétisons les résultats obtenus par les différentes configurations sur le corpus MIX annoté par T. Nakamura et S. Voyatzi. La mesure U_1 correspond à la F-mesure sans tenir compte des étiquettes (uniquement la segmentation). La mesure F_1 est la F-mesure tenant compte à la fois de la segmentation et de l'étiquetage. Nous constatons que lorsque l'analyseur CRF est guidé par POM, les performances grimpent de manière significative (plus de 7 points). La deuxième méthode de combinaison (CRF-adapté) est légèrement meilleure que la première (FUSION).

Chapitre 3

Analyse profonde

Dans cette section, nous continuons l'exploration de l'intégration de ressources lexicales, cette fois appliquée à l'analyse syntaxique profonde, dans deux cadres différents. Dans un premier temps, nous avons cherché à intégrer la reconnaissance des expressions multi-mots dans un processus d'analyse syntaxique probabiliste. Cet aspect est, la plupart du temps, négligé dans la littérature, les études empiriques sur l'analyse syntaxique supposant le plus souvent que celles-ci sont parfaitement reconnues. Nous avons, entre autres, montré l'intérêt d'incorporer des lexiques pour cette tâche. Ensuite, nous avons été amené à travers l'encadrement de la thèse d'A. Sigogne (2008-2012 ?), à travailler sur l'exploitation de lexiques syntaxiques dans des analyseurs probabilistes : analyseurs en constituants basés sur des grammaires PCFG, réordonnanceurs et analyseurs en dépendance.

3.1 Reconnaissance des expressions multi-mots et analyse syntaxique

3.1.1 Etat-de-l'art

L'intégration des expressions multi-mots (EMM) dans des applications réelles comme la traduction automatique ou l'extraction d'information est cruciale car de telles expressions ont la particularité de contenir un certain degré de figement. En particulier, elles forment des unités lexicales complexes qui, si elles sont prises en compte, peuvent non seulement améliorer l'analyse syntaxique, mais aussi faciliter les analyses sémantiques qui en découlent. Elles apparaissent à différents niveaux de l'analyse linguistique : certaines forment des unités lexicales contigues à part entière (ex. les mots composés *cordon bleu*, *par rapport à*, l'entité nommée *San Francisco*), d'autres composent des constituants syntaxiques comme les phrases figées (*N0 prendre le taureau par les cornes* ; *N0 prendre N1 en main*) ou les constructions à verbe support (*N0 donner un avertissement à N1* ; *N0 faire du bruit*).

Les expressions de niveau lexical sont continues et constituent des unités lexicales auxquelles on peut assigner une catégorie grammaticale comme pour les mots simples. L'approche la plus classique est donc de d'abord pré-reconnaitre ces unités et les considérer comme des blocs (i.e. comme des mots simples) en entrée de l'analyseur. Par exemple, [Brun, 1998] réalise une reconnaissance préalable de termes avant d'appliquer une grammaire lexicale fonctionnelle. Néanmoins, la majorité des expériences sur le sujet reposent sur un corpus au sein duquel l'ensemble des EMMs a été parfaitement identifié au préalable. Bien qu'artificielles, ces études ont montré une amélioration des performances d'analyse : par exemple, [Nivre et Nilsson, 2004, Eryigit et al., 2011] pour l'analyse en dépendance et [Arun et Keller, 2005, Hogan et al., 2011] pour l'analyse en constituants. Pour l'analyse en constituants, nous pouvons noter les expériences de [Cafferkey et al., 2007] qui ont essayé de coupler des annotateurs réels de EMMs avec différents types d'analyseurs probabilistes pour l'anglais. Ils ont travaillé sur un corpus de référence non annoté en EMMs. Les EMMs sont reconnues et pré-groupées automatiquement à l'aide de ressources externes et d'un reconnaisseur d'entités nommées. Ils appliquent, ensuite, un analyseur syntaxique et réinsèrent finalement les sous-arbres correspondants aux EMMs pour faire l'évaluation. Ils ont montré des gains faibles mais significatifs. Récemment, [Finkel et Manning, 2009] et [Green et al., 2011] ont proposé d'intégrer les deux tâches dans le même modèle. [Finkel et Manning, 2009] couple analyse syntaxique et reconnaissance des entités nommées dans un modèle discriminant d'analyse syntaxique basé sur les CRF. [Green et al., 2011] a intégré l'identification des mots composés dans la grammaire. Ils ont, en particulier, montré, pour le français, que le meilleur analyseur syntaxique était fondé sur une stratégie non-lexicalisée (PCFG-LA de l'analyseur de Berkeley), bien que l'identification des mots composés soit moins bonne qu'avec un analyseur syntaxique fondé sur une stratégie lexicalisée (avec une grammaire à substitution d'arbres).

Les expressions de niveau syntaxique sont sujettes à des variations syntaxiques, peuvent être discontinues et forment des constituants syntaxiques plutôt que des unités lexicales. Par exemple, des insertions sont possibles : *Max a pris depuis vendredi le taureau par les cordes.* Différentes transformations sont applicables suivant les expressions comme *battre un record* : *Luc a battu un record* ; *C'est un record que Luc a battu.* Il est donc préférable de les reconnaître soit pendant l'analyse, soit après. Nous noterons entre autres l'expérience dans [Roche, 1997] analysant phrases libres et figées au moyen d'un analyseur à états finis [1]. Par ailleurs, [Wehrli et al., 2010] a montré que la reconnaissance de collocations pouvaient se faire en même temps que l'analyse syntaxique. Ils ont montré en particulier que la reconnaissance des collocations permettait de guider l'analyse (et les attachements en particulier). Dans leur outil, les analyses contenant des collocations sont considérées comme plus prioritaires. D'autres études ont montré l'intérêt de reconnaître les EMMs après l'analyse syntaxique comme [Wehrli et al., 2009], qui ont ainsi suivi certaines recommandations de [Heid, 1994]. On

1. Cet analyseur a été implanté comme une cascade de transducteurs se terminant lorsqu'un point fixe a été atteint.

pourrait imaginer utiliser la méthode de [Green et al., 2011] transposable pour tout type d'expressions.

Comme pour tout ce qui concerne les expressions multi-mots, cette frontière entre niveau lexical et syntaxique n'est pas nette. Les mots composés considérés comme des unités lexicales peuvent être discontinues : *à l'insu de*, *à l'insu justement de*. Certaines expressions verbales ont tendance à ne pas contenir d'insertions, même si certaines sont possibles : *faire face* => Je *fais (difficilement+depuis janvier) face*. On a donc envie de les considérer comme des unités lexicales comme c'est le cas dans le corpus annoté de Paris 7 (FTB) [Abeillé et al., 2003]. Certains ont donc mis en place des approches globales et suggérent de pré-reconnaître le plus possible d'expressions multi-mots avant analyse syntaxique ; ceci afin de réduire la taille des textes en terme de mots et donc leur combinatoire pour l'analyse syntaxique. En particulier, [Gross et Senellart, 1998] ont mis au point une stratégie de cascades de transducteurs avec deux types d'opérations : substitution et permutation. L'opération de substitution permet de transformer une séquence de mots non compositionnelle en une seule unité lexicale (ex. *fait divers* → *fait_ divers*). L'opération de permutation permet de rendre continues certaines séquences discontinues. Par exemple, la négation : *ne X pas* → *ne_ pas X* ; ou des expressions figées : *prendre Y en compte* → *prendre_ en_ compte Y*. Les résultats de chacune des étapes sont marqués. Ainsi une phrase telle que *Luc n'a pas pu prendre ce fait divers en compte* serait analysée de la manière suivante :

Luc {ne_ pas/ADV+Neg a_ pu/Vasp prendre_ en_ compte/V}/V ce fait_ divers/NC.

3.1.2 Mots composés et analyse syntaxique

Dans cette sous-section, nous nous limitons aux mots composés. Notre définition est celle du corpus arboré de Paris 7 (FTB), le corpus qui va nous servir pour l'apprentissage et l'évaluation de nos analyseurs. Nous présentons les différentes approches empririques que nous avons testées, en collaboration avec J. Le Roux, A. Sigogne et P. Watrin. Notre but est d'obtenir la meilleure qualité possible d'analyse globale tout en optimisant la qualité d'annotation des mots composés. Nous avons commencé par l'analyse en constituants en utilisant les analyseurs à stratégie non lexicalisée telles ques l'analyseur de Berkeley [Petrov et al., 2006] et l'analyseur LORG [Attia et al., 2010], qui sont *état-de-l'art* pour le français. Ces analyseurs sont basés sur des grammaires PCFG-LA apprises sur le corpus arboré de Paris 7 dans sa version la plus récente (juin 2010) avec tous les mots composés marqués. Nous nous basons sur nos articles publiés [Constant et al., 2012b, Constant et al., 2012a], ainsi que sur un article de revue en cours de soumission [Constant et al., subm]. Cette thématique de recherche est la plus récente (début il y a un an), mais elle a rapidement donné des résultats très intéressants. Ceci peut notamment s'expliquer par le fait que peu d'études ont déjà traité ce sujet. Ainsi, dans cette sous-section, nous décrivons non seulement les stratégies mises en oeuvre et évaluées, mais aussi les différentes perspectives à court et moyen terme que nous envisageons. Nous montrons, entre autres,

l'intérêt d'exploiter des lexiques de mots composés pour cette tâche.

Nous avons d'abord testé la stratégie consistant à reconnaître les mots composés pendant l'analyse syntaxique en incluant leur reconnaissance dans la grammaire. Nous verrons que modifier le schéma d'annotation des mots composés permet d'améliorer leur identification. Nous avons ensuite expérimenté l'approche classique de pré-identification des mots composés dans un contexte réaliste. Contrairement aux travaux dans [Arun et Keller, 2005, Candito et Crabbé, 2009], on ne considérera pas une segmentation lexicale parfaite. On a, en particulier, utilisé un reconnaisseur basé sur les CRF linéaires. Nous verrons ensuite qu'il est possible d'appliquer, à la suite de l'analyseur, un réordonnanceur discriminant intégrant à la fois des traits généraux comme dans [Charniak et Johnson, 2005] et des traits spécifiques aux noms composés. Enfin, nous présenterons une solution prometteuse qui consiste à combiner la méthode par pré-identification des mots composés et la méthode par grammaire intégrant la reconnaissance des mots composés.

Reconnaissance des mots composés dans la grammaire

[Green et al., 2011] ont proposé d'inclure l'identification des mots composés dans la grammaire de l'analyseur. Pour cela, ils utilisent un schéma d'annotation particulier pour les mots composés qui sont annotés par un noeud non-terminal spécifique MWX. Chacun de ses composants est une feuille (token) et son noeud pré-terminal est rattaché au noeud MWX. X correspond à la catégorie grammaticale du mot composé. La figure 3.1 illustre ce schéma d'annotation avec le nom composé *trou noir*. A titre d'exemple, l'analyseur LORG avec une grammaire apprise sur le FTB utilisant un tel schéma d'annotation atteint des performances de l'ordre de 83% en terme de F-mesure sur le corpus d'évaluation. Lorsque l'on considère que les mots composés sont reconnus parfaitement, les performances grimpent à environ 86%, ce qui montre que la reconnaissance des mots composés ne doit pas être négligée. En terme d'identification des mots composés, les résultats sont relativement moyens : de l'ordre de 70% avec le jeu d'étiquettes optimisé pour l'analyse syntaxique [Crabbé et Candito, 2008].

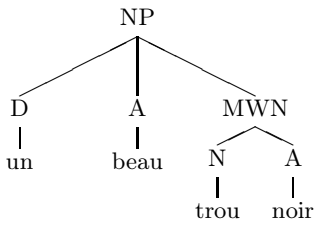


FIGURE 3.1 – Sous-arbre correspondant au mot composé *trou noir*

L'inconvénient de cette annotation est qu'elle ne permet pas de distinguer au niveau lexical (i.e. au niveau des étiquettes grammaticales) si les mots font

partie d'un mot composé. Nous proposons de rajouter un symbole ($^+$) à la catégorie de chaque mot à l'intérieur d'un mot composé. Ainsi, bien que cette annotation augmente le jeu d'étiquettes pré-terminales, elle permet de mieux guider la reconnaissance des mots composés. On peut aller plus loin dans cette stratégie de raffinement de l'étiquetage morphosyntaxique des composants des mots composés en rajoutant à l'étiquette la position relative du composant dans l'unité multi-mots : B si le mot est au début de l'unité, I s'il se trouve aux positions restantes. Ainsi, l'exemple précédent est annoté comme dans les figures 3.2 et 3.3.

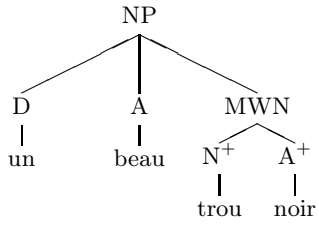


FIGURE 3.2 – Première variante de la représentation du mot composé *trou noir*

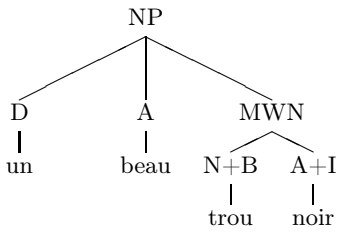


FIGURE 3.3 – Deuxième variante de la représentation du mot composé *trou noir*

Sur le corpus d'évaluation du FTB, nous observons que ces deux variantes obtiennent des résultats comparables. L'analyse syntaxique globale ne s'améliore pas par rapport à l'analyseur appris sur le corpus d'apprentissage avec le schéma d'annotation décrit dans [Green et al., 2011] (notre *baseline*). Par contre, l'identification des mots composés s'améliore de manière significative : entre 1 et 2 points de F-mesure. Ainsi, ce schéma d'annotation n'est utile que si l'on souhaite améliorer la reconnaissance des mots composés.

Pré-identification des mots composés

La reconnaissance de mots composés peut être vue comme une tâche d'annotation séquentielle si l'on utilise le schéma BIO [Ramshaw et Marcus, 1995]. Ceci implique une limitation théorique : les mots composés doivent être continus. Ce

schéma est donc théoriquement plus faible que celui proposé par [Green et al., 2011] qui intègre les mots composés dans la grammaire et autorise des unités polylexicales discontinues. Cependant, en pratique, les mots composés sont très très rarement discontinus et dans la majorité des cas, la discontinuité est due à l'insertion d'un simple modifieur qui peut être incorporé dans la séquence figée : *à court terme*, *à très court terme*. Nous avons proposé d'associer les composants simples des unités polylexicales à une étiquette de la forme CAT+X où CAT est la catégorie grammaticale du mot composé et X détermine la position relative du token dans le mot composé (soit B pour le début – Beginning–, soit I pour les autres positions –Inside–). Les mots simples sont étiquetés O (outside) :

Jean/O observe/O un/O trou/N+B noir/N+I

Ce schéma d'annotation ressemble à celui proposé pour la tâche de segmentation et d'étiquetage [Constant et Tellier, 2012] (cf. section 2.2), à l'exception que les mots simples ne sont pas étiquetés par leur catégorie grammaticale. Les mots composés sont alors regroupés en un seul token. L'analyse syntaxique produit l'arbre suivant :

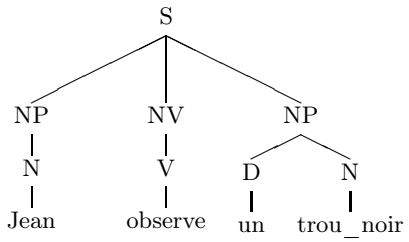


FIGURE 3.4 – Analyse syntaxique par la méthode de pré-identification des mots composés

Pour cette tâche d'annotation des mots composés, nous utilisons, comme dans la section 2.2, le modèle des champs aléatoires markoviens linéaires. Nous y intégrons les mêmes traits. Nous avons également rajoutés des traits dépendants de l'étiquetage grammatical des mots graphiques (i.e tokens). Les étiquettes sont prédites par un simple étiqueteur grammatical (pas de segmentation lexicale). On a aussi rajouté des traits dépendants d'un lexique de collocations nominales [2] candidates calculées à la volée dans le FTB par le système décrit dans [Watrin et François, 2011]. Ces collocations sont notamment associées à leur score d'association (log-vraisemblance). De part leur définition, il paraît naturel que des traits liés à ce lexique peuvent servir d'indice pour repérer des noms composés.

Pour l'évaluation, nous avons ramené les arbres générés par l'analyseur à des arbres comparables avec ceux produits par la méthode précédente. Pour cela,

2. de type statistique.

nous avons reconstruit les sous-arbres associés aux mots composés en déliant ces derniers et en prédisant l'étiquette grammaticale de chacun de leurs composants simples à l'aide d'un étiqueteur. La stratégie par pré-identification obtient des résultats un peu mitigés pour l'analyse syntaxique : nous observons des gains significatifs sur le corpus de développement (environ 0,5%) ; par contre, les différences restent non significatives sur le corpus d'évaluation. En revanche, la reconnaissance des mots composés est améliorée de 7 points par rapport à notre analyseur *baseline*.

On note que l'utilisation de traits basés sur les lexiques externes augmente de manière significative la reconnaissance des mots composés (entre 2 et 4 points). Au niveau de l'analyse globale, les résulats sont plus mitigés, même si l'on constate une tendance à l'amélioration des performances : amélioration forte sur le corpus de développement (+0.5) et pas d'effet sur le corpus d'évaluation. On remarque que les traits purement *endogènes*[3] n'apporte rien au niveau du parenthésage global. Ils améliorent un peu la reconnaissance des mots composés par rapport à la *baseline*, mais sont comparables aux résultats obtenus en utilisant les deux variantes de schéma d'annotation. L'ajout de traits liés au lexique de collocations est une déception. Les gains en terme de reconnaissance des mots composés sont faibles voire nuls.

A. Sigogne a réalisé des expériences similaires dans sa thèse sur le corpus FTB-UC où les noms composés de structure régulière (ex. N+A ou N+P+N) ont été déliés. Il observe des gains significatifs pour cette méthode : +0.7 point pour l'analyse globale, +5 points pour l'identification des mots composés. Nous en concluons que ce sont les noms composés qui troublent l'approche par pré-identification. Cela n'est pas vraiment une surprise, sachant que les noms composés ne sont pas très bien reconnus (entre 3 et 5 points de moins par rapport à la moyenne de l'ensemble des mots composés). Il y a plusieurs explications. Tout d'abord, il y a beaucoup d'incohérences dans l'annotation des mots composés dans le FTB. Par ailleurs, leur structure syntaxique étant très régulière, le reconnaisseur ne peut que se baser sur des distributions lexicales.

Pour terminer, [Arun et Keller, 2005, Nivre et Nilsson, 2004] montrent, dans un cadre idéal, que pré-reconnaitre correctement les mots composés améliore significativement l'analyse syntaxique. Nos études empiriques dans un cadre réaliste nuancent leurs conclusions. En effet, on observe qu'une mauvaise reconnaissance des mots composés provoque des effets de bords sur l'analyse syntaxique globale : les erreurs de reconnaissance se propagent aux constituants supérieurs.

Utilisation d'un réordonnanceur

L'un des problèmes de la méthode de reconnaissance des mots composés intégrée directement dans la grammaire est qu'elle manque d'informations provenant de lexiques de mots composés qui permettrait de mieux guider l'analyseur. Une

3. Les traits endogènes sont des traits calculés exclusivement à l'aide du corpus d'apprentissage ou d'outils appris exclusivement à l'aide de ce corpus.

solution est d'utiliser un rédordonnanceur à la [Collins, 2000, Charniak et Johnson, 2005] se fondant sur des modèles discriminants autorisant l'intégration de traits provenant de tels lexiques. Nous avons donc évalué l'intégration d'un réordonnanceur après l'analyseur syntaxique. Comme dans [Charniak et Johnson, 2005], le celui-ci se base sur un modèle maximum d'entropie. Dans un premier temps, dans [Constant et al., 2012a], nous avons appliqué un modèle incorporant uniquement des traits [4] dédiés aux mots composés relativement comparables avec ceux utilisés dans le reconnaisseur basé sur CRF. Dans [Constant et al., 2012b], nous avons ensuite comparé avec un modèle intégrant aussi les traits généraux [5] décrits dans [Charniak et Johnson, 2005] ou [Collins, 2000]. Pour chaque expérience, le réordonnanceur prenait, en entrée, les 50 meilleures analyses. Par rapport à l'analyseur *baseline*, on constate que l'introduction de traits généraux ne favorise pas la reconnaissance des mots composés (gain limité à 1 point), mais est très pertinente pour l'analyse syntaxique globale (un gain de 0.9 points). Les traits spécifiques aux mots composés améliorent nettement la reconnaissance des mots composés (+5 points) et de manière statisfaisante l'analyse globale (+0.5 points). Par contre, ils sont rendus caducs lorsqu'ils sont combinés avec les traits généraux. Si l'on fait un réordonnancement après l'analyseur utilisant la méthode par pré-identification, on observe des scores relativement comparables par rapport à la méthode *baseline* réordonnancée.

Cette stratégie de réordonnancement obtient des résultats mitigés, mais demande à être approfondie. Le premier point à résoudre concerne la combinaison entre traits généraux et traits dédiés aux mots composés qui ne semblent pas forcément très compatibles. Une méthode à évaluer consiste à appliquer en séquence deux réordonnanceurs : le premier dédié aux mots composés, l'autre dédié à l'analyse générale. La stratégie de réordonnancement est également prometteuse car elle peut être appliquée à tous types d'expressions multi-mots.

Combinaison avec utilisation de treillis

L'un des gros problèmes des analyseurs syntaxiques probabilistes est le traitement des mots inconnus (i.e. absent du corpus d'apprentissage). Un moyen classique d'aider à régler ce problème est d'appliquer un étiqueteur morphosyntaxique puis d'alimenter l'analyseur syntaxique en séquences de mots étiquetés. En effet, les meilleurs étiqueteurs grammaticaux discriminants améliorent l'étiquetage des mots inconnus par rapport à celui produit par un analyseur syntaxique. La différence est d'autant meilleure que l'on intègre des lexiques. Dans notre cas, la reconnaissance de mots composés pose aussi problème. Un reconnaisseur discriminant peut améliorer les résultats de l'identification (gain de 3-5 points). Dans les deux cas (mots inconnus et mots composés), l'intégration de lexiques joue un rôle prépondérant. Cependant, on a vu que la reconnaissance des mots composés par CRF obtenait des résultats moyens (inférieurs à 80%). Il semble donc intéressant de proposer à l'analyseur une entrée ambigue en gardant

4. Nous avons incorporé à la fois des traits endogènes et des traits exogènes.

5. Nous avons utilisé les patrons suivants : *Rule*, *Word*, *Heavy*, *HeadTree*, *Bigrams*, *Trigrams*, *Edges*, *WordEdges*, *Heads*, *WProj*, *NGramTree* et *Score*.

les n meilleures analyses d'un segmenteur étiqueteur. L'analyseur choisira alors les chemins les plus pertinents pour lui. En collaboration avec J. Leroux, nous avons testé cette approche sur les deux types d'analyseurs que nous avons proposés : (a) analyseur avec pré-identification des mots composés et (b) analyseur avec grammaire incluant la reconnaissance des mots composés. Pour (a), nous avons simplement appliqué un segmenteur étiqueteur générant l'automate correspondant aux n meilleures analyses (segmentation et étiquetage). La figure 3.5 illustre l'automate des 4 meilleures analyses pour la phrase *Jean boit de l'eau de vie*.

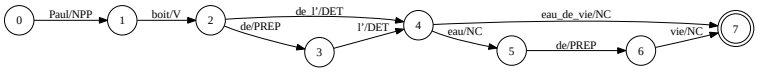


FIGURE 3.5 – Exemple de l'automate des 4 meilleures analyses pour la phrase *Jean boit de l'eau de vie*

Pour (b), nous avons appliqué la même procédure que (a), puis nous avons délié les analyses polylexicales et associé à chacun des composants simples son étiquette grammaticale produite par un étiqueteur. A chacune des étiquettes des composants simples des mots composés, nous ajoutons une marque spécifique (variante 1 du schéma d'annotation des mots composés). La figure 3.6 illustre l'automate décomposé des 4 meilleures analyses pour la phrase *Jean boit de l'eau de vie*.

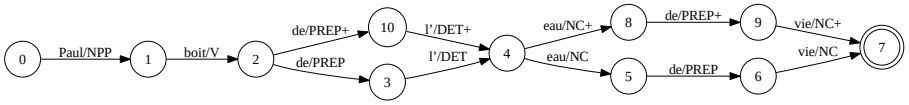


FIGURE 3.6 – Exemple de l'automate décomposé des 4 meilleures analyses pour la phrase *Jean boit de l'eau de vie*.

Ces automates sont donnés en entrée des deux types d'analyseurs. Nous donnons l'évolution de la F-mesure en fonction de n dans la figure 3.7 pour les deux analyseurs. Ces résultats sont décevants. Pour la méthode (a), on s'aperçoit que les meilleurs résultats sont obtenus pour $n = 1$. Ceci s'explique par le fait que l'analyseur a tendance à choisir le chemin le plus court dans le treillis. Pour (b), on otient les meilleurs scores pour $n = 2$, mais avec un score très proche de celui pour $n = 1$.

Ce comportement empirique décevant ne doit cependant pas voiler un aspect encourageant. En effet, si l'on analyse l'oracle, on constate que l'application d'un réordonnanceur pourrait être très prometteur. Dans cette expérience, on passe en entrée de l'analyseur la segmentation (parmi les n proposées) la meilleure par

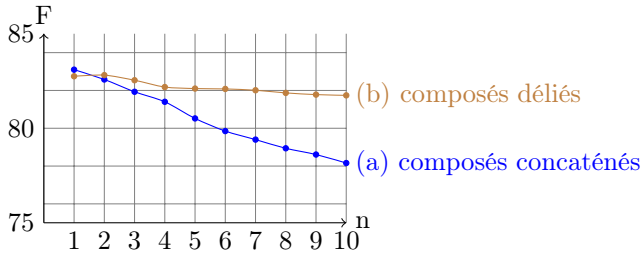


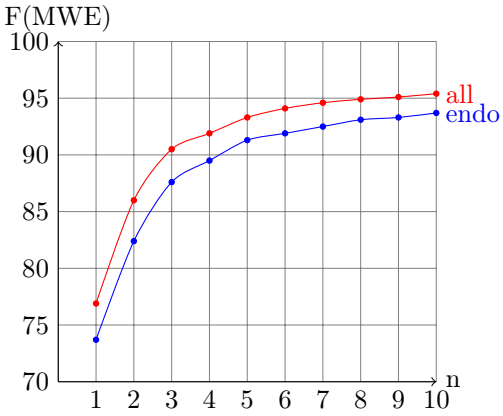
FIGURE 3.7 – Evoluation de la F-mesure en fonction de n

rapport au corpus de référence. Les résultats donnés dans la figure 3.8 montrent que l'on arrive très rapidement à une saturation. Dès $n = 5$, on obtient une qualité d'analyse très satisfaisante, par rapport à ce que l'on peut attendre quand la segmentation lexicale est parfaite.

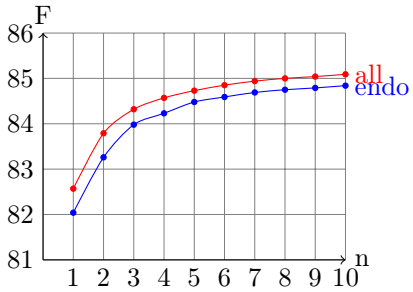
3.1.3 Vers une approche globale ?

Dans la sous-section précédente, nous avons montré quelques stratégies pour intégrer la reconnaissance des mots composés dans un processus d'analyse syntaxique. Dans cette partie, nous envisageons cette problématique dans le cadre général de toutes les expressions multi-mots. D'un point de vue théorique, l'intégration des EMM dans un processus d'analyse syntaxique a déjà fait l'objet de plusieurs études : par exemple, dans une grammaire syntagmatique guidée par les têtes [Copestake et al., 2002] ou dans une grammaire d'arbres adjoints [Abeillé, 1993, Schuler et Joshi, 2011], etc. D'un point de vue empirique, nous avons vu que l'on pouvait envisager la reconnaissance des expressions multi-mots à trois endroits du processus : avant, pendant ou après l'analyse syntaxique.

[Gross et Senellart, 1998] suggèrent de réaliser une pré-reconnaissance séquentielle de l'ensemble des expressions au moyen d'une cascade de transducteurs sous forme de grammaires locales. Cette stratégie, fondée sur des ressources lexicales à grande échelle, a l'avantage de réduire le nombre de "mots" du texte et donc de réduire la complexité combinatoire pour l'analyse syntaxique. Telle qu'elle est présentée, cette approche est séduisante. Cependant, elle a quelques défauts. Tout d'abord, les expressions multi-mots peuvent connaître des variations lexicales et syntaxiques. Pour résoudre en partie ce problème, [Gross et Senellart, 1998] ont proposé une méthode semi-automatique permettant de rendre moins couteuse la construction des grammaires locales. Par contre, il est utile de se poser la question de la pertinence de cette approche pour les collocations (de type lexicographique), cf. [Wehrli et al., 2010]. Par ailleurs, la discontinuité des expressions pose problème. Ce problème est cependant atténué par le fait que les segments discontinus sont souvent proches (quelques



(a) Evolution de l'oracle pour la reconnaissance des mots composés



(b) Analyse de la segmentation "oracle"

FIGURE 3.8 – Les scores "Oracle"

mots tout au plus), ce qui rend leur reconnaissance par grammaire locale moins problématique que dans le cas général des phrases libres. Par ailleurs, ils ont une approche purement non-contextuelle qui ne permet pas de régler certaines ambiguités grammaticales. Par exemple, les séquences ambigues *en fait* et *de la* ne peuvent être résolues sans contexte. Utiliser un modèle discriminant intégrant des traits provenant des grammaires locales pourraient aider à résoudre le problème. Restent des ambiguités de niveau sémantique (*casser sa pipe* = 'casser sa pipe' ou 'mourir') qui ne peuvent être résolues sans connaitre le contexte lexical qui pourrait être intégré dans le modèle discriminant. Enfin, cette

stratégie fait augmenter le nombre de "mots" distincts. Cela cause des problèmes de dispersion lexicale pour l'apprentissage de l'analyseur syntaxique. Ceci pourrait être en partie résolu en remplaçant ces "mots" par des classes plus générales comme les lemmes et des classes apprises de manière non-supervisée, cf. [Candito et Crabbé, 2009]. Il faudrait mettre au point des méthodes de classification spécialisées pour les expressions multi-mots. Bref, il nous semble que cette stratégie mériterait d'être développée et évaluée en tenant compte des différentes améliorations proposées.

Par ailleurs, la méthode de réordonnancement décrite dans la section précédente peut être facilement adaptée à tous types d'expressions multi-mots. Elle a l'avantage de pouvoir intégrer des traits calculés à partir de vastes ressources lexicales. Son autre avantage est qu'elle est intimement liée à l'analyse syntaxique, ce qui permet aux deux tâches de s'aider mutuellement. Les ressources lexicales utilisées pour calculer certains traits du modèle pourraient être appliquées de différentes manières : (a) des heuristiques ; (b) reconnaissance séquentielle par transducteurs ; (c) annotation des noeuds de l'arbre par des modèles discriminants en s'inspirant de [Moreau et Tellier, 2009]. Par exemple, la méthode (b) consisterait à appliquer des transducteurs au moyen d'une procédure permettant de tenir compte des différents niveaux de l'arbre syntaxique. Ainsi, l'expression *prendre en compte*, pourrait être reconnue par un patron du type *NP <prendre> NP en compte* dans l'arbre suivant :

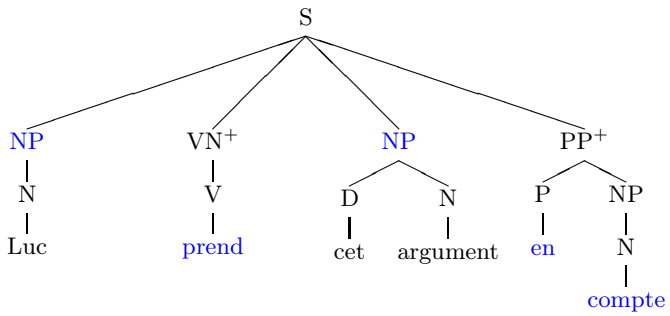


FIGURE 3.9 – Arbre correspondant à l'expression *prendre en compte*.

Ces deux stratégies mériteraient d'être étudiées plus en profondeur. C'est d'ailleurs notre objectif pour les prochaines années.

3.2 Exploitation de lexiques syntaxiques

Dans le cadre de la thèse d'A. Sigogne [Sigogne, 2012], nous avons mis au point différentes stratégies pour intégrer des informations provenant de lexiques syntaxiques dans différents analyseurs syntaxiques probabilistes : (a) des analyseurs se fondant sur des grammaires hors contextes probabilistes PCFG ; (b)

des analyseurs se basant sur des modèles probabilistes discriminants (réordonnanceurs et analyseurs en dépendance). Notre motivation principale était de limiter le problème de la dispersion lexicale en incluant des informations syntaxiques permettant de mieux guider l'analyseur. Dans ces études, une fois n'est pas coutume, nous supposons que les mots composés ont été parfaitement reconnus.

3.2.1 Stratégies

Il existe trois grandes stratégies pour améliorer les analyseurs syntaxiques : (a) modifier les symboles non-terminaux des grammaires ; (b) modifier les mots ; (c) optimiser le jeu de traits dans le cadre des modèles discriminants. Pour chacune de ces trois stratégies, nous avons élaboré différentes méthodes pour intégrer des données provenant de lexiques syntaxiques.

Modification des symboles non-terminaux La littérature regorge d'études sur l'optimisation des symboles non-terminaux. Tout d'abord, certains ont proposé de lexicaliser les grammaires en annotant les noeuds non-terminaux par leur tête lexicale, e.g. [Charniak, 1997, Collins, 2003]. D'autres ont proposé d'atténuer le problème de l'indépendance des hypothèses entre les règles, en introduisant au niveau des noeuds non-terminaux soit des informations provenant du contexte comme dans [Johnson, 1998, Klein et Manning, 2003], soit des symboles latents (cachés) calculés automatiquement comme dans [Matsuzaki et al., 2005, Petrov et al., 2006, Petrov, 2010]. Les grammaires avec annotations latentes, appelées PCFG-LA, obtiennent actuellement des résultats *état-de-l'art*, et, en particulier, pour le français [Seddah et al., 2009, Le Roux et al., 2011]. Dans ce cadre, [Crabbé et Candito, 2008] ont optimisé, de façon *ad hoc*, le jeu d'étiquettes morphosyntaxiques pour le français en utilisant certains traits morphologiques et syntaxiques. [Deoskar, 2008] a mis au point une stratégie consistant à incorporer dans les PCFGs des traits de valence calculés à partir de corpus. Il observe de légers gains en performance.

Dans le même esprit, nous avons essayé d'améliorer le jeu de symboles préterminaux en intégrant des informations provenant de lexiques syntaxiques [Sigogne et al., 2011b]. Pour nos études, nous ne nous sommes intéressés qu'aux verbes et aux noms prédicatifs. Pour cela, nous avons modifié les noeuds préterminaux du corpus d'apprentissage. Pour tous les éléments concernés, la catégorie préterminale CAT est transformée en CAT+CLASSE, où CLASSE est un identifiant de classe calculé à partir d'un lexique syntaxique. Pour les éléments non concernés, elle ne change pas.

Modification des mots Quelques études récentes ont cherché à améliorer les grammaires en modifiant les symboles terminaux. En particulier, elles ont montré l'intérêt de remplacer les mots par des classes plus générales afin de réduire la complexité lexicale. La classe associée à un mot peut être son lemme, une forme

désinfléchie[6] [Candito et Crabbé, 2009], une classe calculée automatiquement avec un algorithme de classification non supervisée (hiérarchique) dans un corpus brut [Candito et Crabbé, 2009], une classe provenant d'un réseau sémantique ou d'un thesaurus [Agirre et al., 2008], etc. Ces modifications sont réalisées par des prétraitements sur les corpus d'apprentissage et d'évaluation. L'idée est de modifier chaque mot par un identifiant de classe et son étiquette grammaticale. Ceci implique l'utilisation d'un étiqueteur grammatical pour prédire l'étiquette morphosyntaxique lors de la phase d'analyse. Cette méthode est aussi applicable sur les modèles discriminants. Dans [Sigogne et al., 2011a, Sigogne et Constant, 2012] et dans sa thèse [Sigogne, 2012], A. Sigogne a repris cette idée appliquée à des classes de mots calculées à partir de lexiques syntaxiques.

Ajout de traits Les modèles discriminants d'analyse syntaxique sont basés sur des traits calculés à partir entre autres (a) des mots et des étiquettes morphosyntaxiques dans le cas des analyseurs en dépendance [Nivre et al., 2006] ; (b) des arbres des n meilleures analyses dans le cas des réordonnanceurs [Collins, 2000, Charniak et Johnson, 2005]. Dans le cadre de l'analyse en dépendance, plusieurs expériences récentes [Koo et al., 2008, Suzuki et al., 2009] ont montré que l'utilisation de classes pertinentes de mots, intégrées sous la forme de traits au modèle, pouvait participer à l'amélioration générale des performances. Il est aussi possible de rajouter des traits liés aux affinités lexicales [Bansal et Klein, 2011, Mirroshandel et al., 2012]. Dans sa thèse, A. Sigogne a essayé d'exploiter des lexiques syntaxiques comme sources de nouveaux traits. En particulier, ces traits correspondent à des classes de mots calculées à partir de lexiques syntaxiques.

Calcul des classes syntaxiques Nous avons mis au point deux méthodes de calcul des classes syntaxiques qui ont été utilisées dans nos expériences. Avec la première méthode, nous partons d'une classification hiérarchique *ad hoc* provenant des tables du lexique-grammaire [Sigogne et al., 2011b]. Les tables du lexique-grammaire forment un ensemble de classes [Gross, 1975, Gross, 1994]. Chaque classe (représentée par une table) regroupe un ensemble d'éléments prédicatifs d'une même catégorie grammaticale partageant un certain nombre de propriétés syntaxiques. Dans le cadre de [Sigogne et al., 2011b], nous en avons dérivé une classification hiérarchique arborescente : chaque noeud de l'arbre correspond à une classe et tous ses enfants partagent un certain nombre de propriétés. Le niveau terminal (niveau 0) correspond aux classes associées aux tables. Pour les verbes, il existe, au total, 4 niveaux : niveau 0 (67 classes), niveau 1 (13 classes), niveau 2 (10 classes), niveau 3 (4 classes). Etant donné un niveau dans la hiérarchie, un élément prédicatif appartient à une ou plusieurs classes, s'il existe dans le lexique-grammaire. L'arbre de la figure 3.10 illustre un extrait de cette classification. Ici, les classes 4, 6 et 12 du lexique-grammaire – cf. [Gross, 1975] – sont regroupées dans une classe QTD2 (transitifs directs à

6. La forme désinfléchie d'un mot correspond à ce mot auquel on a enlevé certaines marques morphologiques tout en préservant l'ambiguité grammaticale.

deux arguments pouvant être sous la forme de complétives). Puis, cette classe est elle-même regroupée avec d'autres classes au niveau supérieur de la hiérarchie pour former une classe TD2 (transitifs directs à deux arguments). La deuxième méthode est générale à tous les lexiques syntaxiques que nous avons utilisés : DicoValence [Van den Eynde et Mertens, 2003], Lefff [Sagot, 2010], LGLex [Constant et Tolone, 2010a], LGLex-Lefff [Tolone, 2011], LexSchem [Messiant et al., 2008]. Tous les lexiques (sauf LGLex) ont en commun qu'ils intégrent, pour chaque entrée lexicale, un cadre de sous-catégorisation avec les arguments liés à leurs fonctions syntaxiques (ex. sujet, obj, objde, etc.) et leur production syntagmatique (ex. groupe nominal, complétive, infinitive, etc.). LGLex associe, à chaque entrée lexicale, un ensemble de propriétés (ex. nature des arguments, distribution lexicale des prépositions, constructions syntaxiques, transformations, etc.).

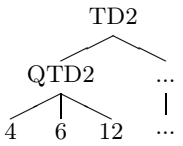


FIGURE 3.10 – Extrait de la classification hiérarchique du lexique-grammaire

Les classes syntaxiques sont calculées de la manière suivante. Une entrée lexicale du lexique est associée à un lemme et à un ensemble de traits sélectionnés manuellement. Un lemme peut correspondre à plusieurs entrées. Ainsi, nous associons à chaque lemme l'union des ensembles de traits correspondant aux entrées auxquelles il appartient. Deux lemmes appartiennent à une même classe s'ils ont strictement le même ensemble de traits. A. Sigogne a sélectionné plusieurs jeux de traits. Pour tous les lexiques sauf LGLex, il a considéré comme traits les fonctions syntaxiques liées aux arguments. Pour LGLex, il a pris pour traits soit les classes de la classification hiérarchique de [Sigogne et al., 2011b], soit les constructions syntaxiques codées. Pour les verbes de LGLex, il a aussi considéré les prépositions liées aux arguments. Pour les noms prédicatifs de LGLex, il a aussi considéré les verbes supports associés. Par exemple, la table 3.1 montre des exemples de classes obtenues sur certains verbes codées dans le Lefff. Le jeu de traits utilisé comprend les fonctions syntaxiques. Dans cette illustration, on notera que l'ensemble de traits associé à *mériter* est l'union de deux ensembles de traits {Suj,Obj} (*Luc mérite cette récompense*) et {Suj,Objde} (*Luc mérite d'être récompensé*).

3.2.2 Résultats expérimentaux

Modification des symboles pré-terminaux L'expérience initiale par modification de symboles pré-terminaux est décrite dans [Sigogne et al., 2011b]. Les classes y sont calculées à partir de la classification hiérarchique *ad hoc*

Verbe	Traits	Classe
abolir	Suj, Obj	1
cibler	Suj, Obj	1
sympathiser	Suj, Obl	2
badiner	Suj, Obl	2
gratifier	Suj, Obj, Objde	3
mériter	Suj, Obj, Objde	3

TABLE 3.1 – Exemples de classes obtenues à partir du Lefff

du lexique-grammaire. Notre stratégie a consisté à réannoter les catégories des verbes et des noms prédicatifs dans le corpus d'apprentissage en fonction de leurs classes et de leurs niveaux d'ambiguité [7]. Etant donné un niveau dans la hiérarchie h et un niveau maximum d'ambiguïté amb, fixés manuellement, on associe à un verbe ou un nom du corpus d'apprentissage sa catégorie grammaticale plus la liste de ses classes de niveau h si son niveau d'ambiguité au niveau h est inférieur ou égal à amb. Nous avons évalué cette méthode sur le FTB-UC par validation croisée en utilisant l'analyseur de Berkeley. Les meilleurs résultats sont obtenus en réannotant uniquement les catégories des verbes n'étant pas ambigus ($amb = 1$) avec l'ensemble des classes de niveau $h = 3$. On observe que les gains obtenus sont faibles mais significatifs (+0.3%) avec validation croisée sur le FTB-UC. La réannotation des noms n'a aucun effet significatif. Ceci peut s'expliquer par la faible couverture du lexique-grammaire pour les noms. Récemment, en répliquant ces expériences avec une nouvelle version de l'analyseur de Berkeley, nous avons observé que cette approche n'avait plus aucun effet. Ceci s'explique par la méthode d'apprentissage de la grammaire qui cherche déjà à optimiser le jeu de symboles non-terminaux. Celle-ci s'améliorant, elle a tendance à rendre caduques nos améliorations.

Modification des symboles terminaux Nous avons ensuite réalisé un certain nombre d'expériences consistant à remplacer chaque élément prédicatif par sa catégorie morphosyntaxique et un identifiant de classe syntaxique. Dans un premier temps, nous avons calculé ces classes à partir des tables du lexique-grammaire et de la hiérarchie décrite ci-dessus [Sigogne et al., 2011a]. Puis, nous avons cherché à obtenir ces classes à l'aide des cadres de sous-catégorisation du Lefff [Sigogne et Constant, 2012]. Les expériences décrites dans ces articles montrent de faibles gains (0.5-0.6%) en terme de F-mesure mais significatifs, par validation croisée sur le FTB-UC, avec l'analyseur de Berkeley. Par contre, la méthode de classification non supervisée décrite dans [Candito et Crabbé,

7. Le niveau d'ambiguité d'un verbe ou d'un nom correspond au nombre de classes qui lui sont associées à un certain niveau de la hiérarchie.

2009] bat largement ces méthodes (+1.4%). La combinaison des deux approches n'a aucun effet par rapport à la méthode de [Candito et Crabbé, 2009] seule. Enfin, A. Sigogne a étendu ces expériences à tous les lexiques cités dans la section 1.3.3 et à un analyseur à stratégie lexicalisée, l'analyseur Brown [Charniak, 2000]. Dans sa thèse [Sigogne, 2012], il fait une synthèse des méthodes de calcul de classes et des évaluations dans un même cadre expérimental. Les différentes méthodes ont été testées sur les corpus de développement et d'évaluation de FTB-UC. Tout d'abord, il a évalué séparément l'impact de chaque catégorie (verbes, noms et adjectifs). Ce sont les expériences avec les verbes qui ont les meilleurs résultats. Les gains en terme de F-mesure sont faibles (entre 0.1 et 0.6%) mais ils sont souvent statistiquement significatifs, quels que soient le lexique utilisé et le nombre de classes générées. Par contre, les noms prédicatifs et les adjectifs n'ont aucun effet. On observe cependant que, lorsque l'on utilise les catégories grammaticales de référence, les gains doublent et sont quasiment toujours significatifs, sauf pour les noms. Ceci montre clairement que cette méthode est très sensible à l'étiquetage grammatical. Par ailleurs, la combinaison des catégories grammaticales fait stagner ou dégrade les résultats par rapport à ceux obtenus en travaillant sur les verbes uniquement. De plus, comme dans [Sigogne et al., 2011a, Sigogne et Constant, 2012], ces méthodes ont moins d'effets que celles décrites dans [Candito et Crabbé, 2009]. Leur combinaison ne donne rien non plus. Si l'on compare les analyseurs, on remarque que l'analyseur Brown est plus sensible à nos prétraitements que l'analyseur Berkeley. Ceci peut s'expliquer qu'il utilise une stratégie lexicalisée. Le corpus d'apprentissage étant trop petit, ce sont le plus souvent les probabilités de lissage qui sont utilisées lors de l'application de la grammaire. Remplacer les mots par des classes plus générales a donc un impact plus grand qu'un analyseur à stratégie non lexicalisée qui cherche justement à "généraliser". Plus, cette généralisation sera bonne, moins il aura besoin de nos prétraitements. On notera que les anciennnes versions de Berkeley (1.1 et 1.2) obtiennent des meilleurs scores avec nos prétraitements. L'utilisation d'une version de l'analyseur de Berkeley plus récente diminue l'impact de nos ressources lexicales. On note que pour une évaluation par validation croisée sur FTB-UC, l'effet de l'approche est quasiment nul en terme de F-mesure, sauf si l'on utilise les étiquettes grammaticales de référence (+0.5%). Par contre, en terme de dépendances non typées, les scores sont significativement améliorés (+0.4 pour Berkeley et +0.6 pour Brown).

Traits dans les modèles discriminants Dans sa thèse [Sigogne, 2012], A. Sigogne a également essayé d'exploiter des lexiques syntaxiques dans des modèles discriminants pour l'analyse syntaxique. Il en existe deux types : (a) les réordonnanceurs qui prennent en entrée les n meilleures analyses d'un analyseur lambda et les reclasse en fonctions de traits non-locaux comme dans [Charniak et Johnson, 2005] ; (b) les analyseurs en dépendance comme dans [Nivre et al., 2006, Mcdonald et al., 2005]. Pour les modèles en dépendance, l'impact de la dispersion des données sur ces modèles est moindre que pour les modèles génératifs. Aussi, l'utilisation de regroupements lexicaux n'est pas aussi cruciale

pour ce type d'analyse. Malgré tout, plusieurs expériences récentes [Koo et al., 2008, Suzuki et al., 2009] ont montré que des regroupements pertinents, intégrés sous la forme de traits au modèle, pouvaient participer à l'amélioration générale des performances. Il y a deux approches possibles pour utiliser les lexiques syntaxiques. Soit, comme pour les grammaires PCFG, on remplace les mots par des classes calculées à partir d'un lexique syntaxique, soit on garde les mots tels quels mais on intègre, dans le modèle, des traits "exogènes" correspondant à ces classes. Pour ses expériences, A. Sigogne a utilisé une implantation maison du réordonnanceur de [Charniak et Johnson, 2005] et l'analyseur en dépendance MaLT [Nivre et al., 2006]. Il utilise le lexique LexSchem [Messiant et al., 2008] pour calculer les classes. Dans tous les cas de figure, l'hypothèse que les modèles discriminants souffriraient moins de la dispersion lexicale est vérifiée par nos expériences. En effet, que ce soit pour le réordonnanceur ou pour l'analyseur en dépendance Malt, les différentes stratégies d'intégration de données lexicales et syntaxiques se sont révélées inefficaces car les gains sont nuls ou non-significatifs.

Améliorations possibles Les expériences que nous avons décrites montrent que la méthode par modification de symboles terminaux des grammaires est la meilleure, même si les résultats ne sont pas extraordinaires. On peut donc se demander quelles seraient les améliorations à apporter pour que l'exploitation de lexiques syntaxiques soit utile. Tout d'abord, les résultats montrent clairement que la méthode est très dépendante de la qualité de l'étiquetage grammatical (ou de la lemmatisation). Une solution à explorer est donc d'améliorer l'étiqueteur utilisé, peut-être en tentant d'exploiter automatiquement les erreurs provoquées dans un analyseur. Un autre point d'amélioration serait de plus lier le contenu du lexique syntaxique avec le contenu du texte. Nous avons vu qu'un lemme pouvait appartenir à plusieurs entrées lexicales qui ont des cadres de sous-catégorisation différents (ex. *voler = L'oiseau vole* ou *Luc vole un oiseau à Lea.*). Une solution simple serait de mettre au point un système de levée d'ambiguité en se servant du contexte autour du mot en question. A. Sigogne a notamment développé un petit module se basant sur les prépositions dans le contexte du mot à désambiguiser. Malheureusement, il montre que cela ne change pas les résultats obtenus. Une autre solution serait de faire des études systématiques d'erreurs générées par les analyseurs et dans tous les cas voir quelles propriétés du lexique permettraient de résoudre le problème. Bien que nous nous sommes aidés de ce type d'analyses pour mettre au point nos méthodes, nous avons surtout eu recours à une approche *ad hoc*.

3.2.3 Y a-t-il un avenir pour les lexiques syntaxiques ?

A la vue des résultats obtenus dans les sous-sections précédentes, on est en droit de s'interroger sur l'avenir des lexiques syntaxiques dans le cadre de l'analyse syntaxique probabiliste. Si l'on regarde les résultats de nos méthodes sur les grammaires PCFG (les seules qui donnent des gains positifs et significatifs), on s'aperçoit que le meilleur lexique est LexSchem [Messiant et al., 2008], un lexique qui a été appris automatiquement à partir d'un corpus. Il parait donc logique

de se poser la question de l'utilité des lexiques syntaxiques construits manuellement (ex. lexique-grammaire) pour ce type d'analyseur. Par ailleurs, nous avons aussi montré que les gains obtenus par nos méthodes étaient moins élevés que ceux obtenus par des méthodes totalement automatiques de classification non supervisée ou regroupement de mots [Candito et Crabbé, 2009]. En combinant les deux approches, nous n'avons rien observé de significatif. La question de l'utilité d'un lexique syntaxique dans un analyseur syntaxique probabiliste est donc posée...

Malgré ces résultats négatifs, il nous semble intéressant, sur le court terme, de pousser les expériences dans différentes directions. Tout d'abord, nous pourrions tester les méthodes à l'adaptation de domaines. Un autre point serait de trouver un moyen de combiner correctement le méthodes de [Sigogne et Constant, 2012] et celles dans [Candito et Crabbé, 2009]. Nous pourrions aussi nous baser sur l'exploitation automatique des erreurs des analyseurs pour sélectionner les propriétés intéressantes des lexiques syntaxiques. Il serait aussi intéressant de réitérer nos expériences sur d'autres langues. L'anglais, par exemple, dispose de corpus annotés de grande taille mais également de nombreux lexiques syntaxiques : ComLex [Grishman et al., 1994], NomLex [Macleod et al., 1998], FrameNet [Baker et al., 1998] ou encore VerbNet [Kipper et al., 2000]. Ce dernier est une ressource proche du Lexique-Grammaire car il regroupe les verbes en classes d'après leurs comportements syntaxiques et sémantiques. Par ailleurs, nous pourrions également exploiter les grammaires d'unification comme HPSG, CCG (grammaires catégorielles combinatoires) ou encore LFG. Bien que les analyseurs basés sur ces grammaires ne permettent pas d'obtenir actuellement des performances au niveau de l'*état de l'art*, elles disposent néanmoins d'un formalisme plus adapté à l'intégration de données lexicales et syntaxiques. [Carroll et Fang, 2004] ont, par exemple, réussi à réduire d'environ 5% le nombre d'erreurs d'un analyseur HPSG de l'anglais en utilisant des informations de sous-catégorisation de verbes contenues dans un lexique obtenu de manière automatique.

Il nous semble néanmoins que l'avenir des analyseurs syntaxiques probabilistes réside plus dans l'exploitation des affinités lexicales comme dans [Bansal et Klein, 2011, Mirroshandel et al., 2012], que dans celle de lexiques syntaxiques. Loin de moi l'idée d'enterrer les lexiques syntaxiques pour le TAL ! Les lexiques tels que ceux issus des tables du lexique-grammaire [Gross, 1975] ou le dictionnaire [Dubois et Dubois-Charlier, 1997] seront toujours utiles pour de futures analyses sémantiques car ils peuvent être une source pour aider à lier syntaxe et sémantique. Dans ces lexiques, chaque emploi correspond à un sens. Les arguments syntaxiques peuvent être liés aux arguments sémantiques des prédicats associés : par exemple, les arguments numérotés N0, N1, N2 dans le lexique-grammaire.

Chapitre 4

Applications

Durant ces quelques années dans le monde de la recherche, nous nous sommes aussi intéressé à des aspects plus applicatifs. Tout d'abord, au contact du monde privé, nous avons participé à l'élaboration d'applications industrielles liées à l'extraction d'informations. Ensuite, au contact de linguistes, nous avons développé des applications linguistiques comme le traitement de transcriptions orales pour faciliter des études linguistiques sur les disfluences, ainsi que l'extraction de traductions d'expressions multi-mots pour aider à la construction de dictionnaires bilingues. Dans toutes ces applications, des prétraitements fins à base de ressources lexicales ont été mis en oeuvre afin de permettre la pré-reconnaissance des expressions multi-mots continues telles que les mots composés et les entités nommées.

4.1 Applications industrielles

Avec l'explosion du nombre de documents textuels, il est nécessaire de développer des outils permettant de structurer les documents afin de mieux les indexer et donc de faciliter la recherche d'informations. C'est, dans cette optique, qu'entre 2003 et 2008, nous avons travaillé au contact du monde industriel, principalement sur l'extraction d'unités lexicales pertinentes et sur le développement d'algorithmes pour l'indexation et la classification de documents. Tout d'abord, en 2003-2004, nous avons été employé en tant que chercheur [1] pour l'entreprise Teragram Corporation (Boston, Etats-Unis). A notre retour en France, j'ai continué à garder des liens avec le monde industriel et à développer des applications hybrides pour diverses entreprises (SeniorPlanet, Softissimo, Xeres) ou consortium (projet Outilex).

Dans cette section, nous revenons sur deux projets applicatifs que nous considèrons les plus significatifs scientifiquement : (a) nos travaux à Teragram Cor-

1. Durant cette période, nous étions soumis à une clause de confidentialité nous empêchant de publier nos résultats. C'est d'ailleurs en très grande partie à cause de cette clause que nous avons décidé de rentrer en France dans le but de trouver un poste académique.

poration sur le classement et le regroupement de mots dans des textes (sous-section 4.1.1) ; (b) les travaux d'A. Sigogne que nous avons supervisé pour la compagnie *Xeres* (sous-section 4.1.2). Les applications décrites sont basées sur des prétraitements linguistiques fins incluant la reconnaissance des mots composés et des entités nommées à partir de dictionnaires et de grammaires locales fortement lexicalisées. Cela permet (a) de repérer des unités sémantiques complexes telles que *trou noir* ou *Maison Blanche* qui sont indécomposables ; (b) de filtrer certains mots simples pleins faisant partie de mots vides composés comme *à peine*, *en effet* ou *est en train de*. Dans une dernière partie, nous survolerons les autres outils sur lesquels nous avons travaillé.

4.1.1 Classement et regroupement de mots

Cette section reprend une partie des travaux que nous avons réalisés à Teragram Corporation en 2003-2004. L'entreprise était spécialisée dans l'indexation intelligente de documents pour la recherche fine d'informations dans des bases de documents textuels. Nos travaux avaient pour but de mettre au point des méthodes d'extraction d'informations dans des documents afin d'aider à structurer et indexer ces derniers. Les différentes informations extraites étaient les noms de personnes, les noms de lieux, les noms d'organisation, les mots-clés ou les thèmes abordés. Nous avons utilisé des algorithmes basés sur des graphes, qui, à l'époque de mon séjour à Teragram, commençaient à être exploités sérieusement pour la recherche d'information [Brin et Page, 1998, Haveliwala, 2003], le résumé automatique [Erkan et Radev, 2004] ou la levée d'ambiguïté [Veronis, 2003]. Ces techniques basées sur des graphes de mots ont depuis montré de nombreuses applications et forment un domaine à part entière dans la communauté du TAL comme en témoigne l'organisation annuelle de l'atelier international *Workshop on Graph-based Algorithms for Natural Language Processing* depuis 2006 ou le livre [Mihalcea et Radev, 2011].

Dans ce contexte, nous avons travaillé sur deux tâches : (a) regrouper sémantiquement les unités lexicales d'un nouveau document afin d'identifier facilement les thématiques abordées ; (b) ordonner les unités en fonction de leur importance dans le document afin d'identifier les mots-clés. Les deux applications sont basées sur un graphe représentant le degré de proximité (ou similarité) sémantique entre les différentes unités lexicales du texte. Pour la tâche de regroupement, on regroupe les unités les plus proches sémantiquement dans le graphe par des techniques de classification non-supervisée (ou *clustering* en anglais). Pour la sélection des unités les plus importantes, on se base sur le principe récursif du PageRank de Google [Brin et Page, 1998] : plus une page est citée par des pages populaires, plus elle est populaire. Appliqué à notre cas, ce principe devient : plus une unité a des unités importantes sémantiquement proches dans le texte, plus elle est considérée comme importante. Dans le graphe représentant la similarité sémantique, chaque sommet correspond à une unité lexicale. Un arc relie deux unités proches sémantiquement : cette proximité est pondérée par un réel compris entre 0 et 1. La similarité sémantique entre deux unités est calculée en fonction de leurs contextes lexicaux. Deux unités sont considérées comme

proches sémantiquement si elles ont des contextes lexicaux similaires [Manning et Schütze, 1999]. La similarité sémantique est d'ailleurs souvent considérée comme une cooccurrence de deuxième ordre. Dans notre cas, le contexte lexical d'une unité est la phrase dans laquelle elle apparaît. Dans ce cas précis, la similarité sémantique correspond plutôt à une proximité thématique. N'ayant pas publié sur le sujet, je détaille un peu plus que la normale les points abordés.

Construction du graphe

La première phase est de constituer un graphe reliant les unités lexicales proches sémantiquement à partir d'un corpus d'apprentissage (dans notre cas, 3 ans du New York Times). Cette phase est composée de trois sous-étapes : (a) extraction des unités lexicales pertinentes à base de procédures linguistiques ; (b) calcul des K meilleurs coocurrents pour chacune des unités ; (c) constitution du graphe en fonction des listes des meilleurs cooccurents pour chaque unité. Lors du traitement d'un nouveau document (en général, des dépêches ou des articles journalistiques) pour indexation par exemple, le graphe sera réduit aux sommets dont les unités appartiennent au nouveau document. Les documents étant relativement petits, nous avons fait l'hypothèse relativement raisonnable qu'une unité potentiellement ambigue n'a qu'un seul sens tout au long du document dans lequel il apparaît.

Pour l'étape (a), nous avons mis au point un analyseur de surface basé sur des grammaires locales permettant de repérer les mots pleins simples et composés du texte. Dans notre travail, nous nous sommes limités aux noms. Cet analyseur permet tout d'abord de pré-sélectionner des unités complexes telles que *United States* ou *prime minister* qui ne peuvent pas être décomposées. Il permet également de supprimer certains noms simples appartement à des unités grammaticales complexes : par exemple, *of course* est étiqueté comme un adverbe et *a bench of* est reconnu comme un déterminant. Nous donnons ci-dessous un exemple de phrase analysée :

[Tillery] ~~was charged with~~ [first-degree murder] ~~in the~~ [killing] ~~of~~ [Charles Johnson].

Lors de l'étape (b), pour chaque phrase, nous sélectionnons un ensemble d'unités qui seront considérées comme mutuellement cooccurrentes. Ainsi, sur l'ensemble du corpus d'apprentissage, chaque unité lexicale possède une liste propre d'unités coocurrentes dont nous gardons les N plus fréquentes (dans notre cas, $N = 1000$). L'exemple ci-dessous correspond au début de la liste des unités cooccurrentes [2] les plus fréquentes du mot simple *emissions*. A chaque unité cooccurrente, on associe le nombre de fois où elle apparaît dans la même phrase que *emissions* dans le corpus d'apprentissage, i.e. le nombre de cooccurrences.

```
442,extr:emissions
55,extr:mr
```

2. On notera que *emissions* est cooccurrent avec lui-même.

```
52,extr:percent
50,extr:carbon dioxide
40,extr:global warming
35,extr:greenhouse gas
29,extr:companies
28,extr:year
27,extr:carbon emissions
25,extr:administration
24,extr:bush
24,extr:program
24,extr:rules
24,extr:greenhouse gases
23,extr:pollution
23,extr:united states
23,extr:pollutants
23,extr:power plants
22,extr:reductions
22,extr:standards
22,extr:gases
22,extr:cut
21,extr:fuel
21,extr:plan
20,extr:state
...
```

Afin de filtrer et donc affiner cette liste, nous avons utilisé une formule mesurant le degré de cooccurrence entre une unité lexicale u et une de ses unités coocurrentes uc : la F_β-mesure. Cette mesure, très classique pour l'évaluation en TAL avec $\beta = 1$, est rarement utilisée pour notre tâche. Elle inclut un paramètre β et combine précision (p) et rappel (r).

$$F_\beta = \frac{(1 + \beta^2).p.r}{\beta^2 p + r}$$

Dans notre cas, le rappel correspond au nombre total de co-occurrences entre u et uc normalisé par le nombre total d'occurrences de u. La précision correspond au nombre total de cooccurrences entre u et uc normalisé par le nombre total d'occurrences de uc. Pour notre application, le paramètre β est manuellement fixé à 3. Cette mesure permet d'avoir une pondération plus fine des unités cooccurrentes que leur fréquence et donc de se limiter à un nombre plus restreint de cooccurrents par unité lexicale. Parmi les N unités cooccurentes les plus fréquentes pour une unité lexicale donnée, nous gardons les K meilleures en fonction de la nouvelle pondération ($K = 20$ dans notre cas). La liste des 20 meilleures unités cooccurentes de *emissions* est donnée ci-dessous.

```
emissions
--------
```

```
emissions 0.9943757030
carbon dioxide 0.1184553423
global warming 0.0932618326
greenhouse gas 0.0869133350
carbon emissions 0.0674157303
greenhouse gases 0.0590115564
pollutants 0.0554617796
power plants 0.0531055184
gases 0.0529610014
pollution 0.0517668242
reductions 0.0494715539
emissions of carbon dioxide 0.0475356517
kyoto 0.0440960314
sulfur 0.0417075564
fuel 0.0400686892
kyoto protocol 0.0372856078
climate change 0.0364785992
dioxide 0.0348085530
standards 0.0340715503
vehicles 0.0340657469
```

L'étape (c) consiste à construire le graphe à partir des listes calculées précédemment. Chaque unité lexicale du corpus d'apprentisssage correspond à un sommet du graphe. Chaque arc entre deux sommets est pondéré par le degré de similarité sémantique entre les deux unités correspondantes. On suppose qu'un arc existe si les unités sont suffisamment proches sémantiquement, c'est-à-dire si le degré de similarité est supérieur à un seuil donné. Pour calculer la proximité sémantique, nous projetons notre problème dans un espace vectoriel classique. Chaque unité lexicale correspond à un vecteur. Chaque composante est liée à une unité du vocabulaire de la langue et indique le degré de cooccurrence entre les deux unités. Le degré de cooccurrence est la F_β-mesure pour les K meilleures unités cooccurrentes et 0 pour les autres. Pour calculer la similarité entre deux unités (i.e. deux vecteurs), nous avons utilisé des mesures standard entre vecteurs comme le *cosinus* [Manning et Schütze, 1999] ou des formules proches. Le seuil pour déterminer si deux unités lexicales sont liées par un arc est calculé dynamiquement en fonction du contenu du graphe.

Etant donné un nouveau document, nous extrayons ses unités lexicales et nous ne gardons que les sommets dont les unités appartiennent au texte. Nous avons donc un nouveau graphe qui représente la proximité sémantique entre les unités du texte.

Regroupement d'unités lexicales

La tâche de regroupement d'unités lexicales en classes sert à aider à trouver les thématiques abordées dans le texte. Nous nous sommes basé sur une extension de l'algorithme des k-moyennes proposée par [Aggarwal et al., 2004].

Le problème avec cet algorithme est qu'il a besoin que k, le nombre de classes, soit fixé. Or, dans notre cas, le nombre de thématiques abordées n'est pas fixe. Il est donc nécessaire de calculer dynamiquement cette valeur en fonction du texte traité. Notre solution a consisté à réaliser un pré-regroupement simple à partir du graphe de proximité : chaque composante connexe du graphe du texte correspond à une classe d'unités lexicales. Cette solution a l'avantage de pré-calculer k et de fournir un ensemble de k graines plutôt bonnes à l'algorithme des k-moyennes. Nous avons d'ailleurs observé que la classification initiale (les graines) était souvent très proche de la classification finale. On associe un poids aux classes d'unités lexicales calculées afin de mettre en valeur les thèmes les plus importants. Ce poids est la somme du nombre d'occurrences des unités de la classes dans le texte traité.

Il peut être intéressant d'étendre les classes du texte à des unités non présentes dans le texte, ceci afin de généraliser les classes et ainsi améliorer l'indexation des documents. La méthode que nous avons proposée se fonde sur les listes des meilleurs cooccurrents calculées dans la phase de construction du graphe. Soit unk une unité du corpus d'apprentissage absente du texte traité. On rajoute unk à une classe C du texte si unk appartient aux K meilleures unités cooccurrentes d'un certain nombre d'unités de C (défini par un ratio).

Classement d'unités lexicales

Afin de sélectionner les unités lexicales les plus importantes d'un texte, il est nécessaire de les classer selon leur degré d'importance. Pour calculer ce degré d'importance, nous nous sommes basé sur le principe récursif du PageRank de Google [Brin et Page, 1998] : plus une page web est citée par des pages populaires, plus elle est populaire. Appliqué à notre cas, ce principe devient : plus une unité a des unités importantes sémantiquement proches dans le texte, plus elle est considérée comme importante. Ce classement étant général et dépendant du corpus d'apprentissage, nous combinons le degré d'importance calculé par la méthode ci-dessus à la fréquence de l'unité dans le texte traité et la taille de la classe à laquelle elle appartient [3].

Cette pondération des unités lexicales a été utilisée pour extraire des classes lexicales pour le système de classification supervisé de Teragam. Nous avions à notre diposition une collection d'entraînement où chaque document était associé à une catégorie thématique. Pour chaque document d'une catégorie donnée, nous avons extrait les unités les plus importantes à l'aide de la pondération décrite ci-dessus. Nous avons alors concaténé ces listes d'unités importantes dans un seul document auquel nous avons réappliqué la procédure d'extraction des unités les plus importantes. Nous obtenons ainsi une classe d'unités lexicales associée à une catégorie donnée du système. Par exemple, pour la catégorie *Research/Health*, nous obtenons la liste d'unités suivante :

```
disease (0.0234454729579777)
```

3. Pour les unités inconnues (i.e. absentes du corpus d'apprentissage), nous utilisons une pondération particulière.

```
diseases (0.0210760012289433)
researchers (0.0208212528963188)
patients (0.0203428360658205)
prevention (0.0193047704466931)
disease control (0.0185973291231076)
study (0.0176669090828246)
studies (0.0168646624631823)
scientists (0.0168098959721839)
public health (0.0168090032945288)
treatment (0.0167924303559082)
doctors (0.0166153438857456)
health (0.0160392976606167)
drugs (0.0150807990504093)
effects (0.0139549848872999)
treat (0.0139359798244405)
brain (0.0135806003337323)
prescription (0.0135374703005999)
risk (0.013488824565556)
research (0.0130156083169856)
drug (0.0129764604044519)
health care (0.0128510073789425)
humans (0.0119742138513067)
infection (0.0118872373943332)
science (0.0114881835371277)
journal (0.011437916776131)
centers (0.0108089211841234)
health organization (0.0107490253289497)
animals (0.0105310176348728)
institute (0.0101850651391506)
```

4.1.2 Classification temps-réel de documents web

L'information sur le Web évolue continuellement alors que de nouveaux documents sur de nouveaux sujets sont rendus disponibles tous les jours. L'entreprise *Xeres* est spécialiste de la veille documentaire. Tous les jours, cette entreprise reçoit des dizaines de milliers de documents web (blogs, news, forums, etc.) sur divers sujets impliquant leurs clients et est chargée de découvrir les opinions émergentes, les modes et les actualités autour de ces sujets, etc. Le nombre de documents à examiner étant très grand, elle a besoin d'outils permettant d'extraire automatiquement les sujets abordés dans chaque document et de regrouper les documents par sujets. C'est dans ce contexte que Xeres a pris contact avec l'équipe d'informatique linguistique du LIGM. Durant son stage de Master 1 que nous avons encadré d'avril à septembre 2008, A. Sigogne a été chargé de développer un outil de classification non supervisée qui classe les pages web au fur-et-à-mesure de leur arrivée. Afin d'éviter de recalculer de manière itérative tous les regroupements à chaque fois qu'un document arrivait (comme dans

l'algorithme classique des k moyennes), il a fallu réaliser une procédure peu complexe en temps de calcul. La sortie est un ensemble de classes auxquelles ont associe un ensemble de mots clés et les liens hypertexte vers ses documents. Chaque lien est accompagné d'un petit résumé et d'un ensemble de mots-clés. Une classe correspond à un sujet très précis et contient en général très peu de documents. L'article [Sigogne et Constant, 2009] présente l'outil développé par A. Sigogne durant son stage. Il décrit d'abord comment sont représentés les documents puis comment ceux-ci sont affectés à des classes. Sans grande originalité, chaque document est représenté par un vecteur. Chaque composante de ce vecteur est liée à une unité lexicale du vocabulaire des documents et indique le poids de l'unité dans le document. Le première tâche est de détecter les unités pertinentes du texte afin de leur affecter un poids non nul (les autres unités auront un poids nul). La procédure utilisée est la suivante. On extrait d'abord le texte pertinent du document web : on enlève les publicités, images, balises, menus de navigation, liens vers d'autres pages incluant des mini-résumés ... On se base sur des heuristiques basées sur des critères simples : taille du bloc de phrases contigues, longueur des phrases, distance entre les phrases dans le document en terme de balises. Puis, on détecte automatiquement la langue afin d'exploiter les ressources lexicales correspondantes. Cette détection se fonde sur un algorithme identifiant les facteurs interdits d'une langue qui peuvent être extraits à partir d'un petit échantillon de textes. Ensuite, on segmente lexicalement le texte à l'aide des ressources lexicales (dictionnaires de mots composés et grammaires locales d'entités nommées). Le texte est alors étiqueté grammaticalement à l'aide de treetagger [Schmid, 1994]. On ne garde que les noms qui sont alors lemmatisés. Ces noms sont alors filtrés en fonction de leur TF.IDF. Notons que nous avons adapté la mesure du TF.IDF aux noms composés en tenant compte des composants internes afin de favoriser ces unités complexes.

La procédure de classification non-supervisée est une combinaison des méthodes décrites dans [Aggarwal et al., 2004] et [Radev et al., 1999]. Chaque classe contient un ensemble de documents. Elle est aussi représentée par un vecteur dans le même espace que les documents. Le vecteur d'une classe est le centroïde des vecteurs de ses documents. La première étape est de réduire la taille de l'espace vectoriel en ne gardant que les unités lexicales les plus discriminantes pour l'affectation des documents. On utilise la formule du *gini-index* décrite dans [Aggarwal et al., 2004]. Cette mesure sert de critère pour garder ou enlever une unité de l'espace. L'algorithme de classification est classique et ne contient qu'une seule passe : il classe un document à la fois et lui affecte la classe la plus proche en fonction de la mesure du *cosinus* comme dans [Radev et al., 1999]. Si aucune classe n'est assez proche, une nouvelle classe est créée et le document lui est affecté. Quand un document est affecté à une classe, cela peut causer deux types d'effet de bord : (1) la classe peut devenir très similaire à une autre et il est utile de les fusionner, (2) elle peut devenir incohérente et il est nécessaire de la diviser. Dans notre système, la classification est donc affinée au moyen d'opérations de fusion de classes et de division de classes décrites dans [Aggarwal et al., 2004]. Celles-ci ont cependant été optimisées pour satisfaire les contraintes de qualité et de temps de la part de Xeres. Par ailleurs, pour éviter de recalculer le

gini-index à chaque fois que la classification est modifiée, ce qui est coûteux en temps, ce calcul n'est réalisé que tous les 100 documents. Ceci n'a aucun effet sur la qualité du système car l'insertion d'un nouveau document modifie très peu les statistiques pour le calcul du *gini-index*.

Notre outil a été évalué par trois personnes extérieures au projet. Cette évaluation a montré que le nettoyage des pages web était quasiment parfait. Le module de détection de la langue donne de bons résultats pour les trois langues gérées (anglais [94%], espagnol [92%], français [99%]). Les erreurs étaient essentiellement dues à des documents trop courts (documents *YouTube* par exemple). L'évaluation de l'algorithme de classification a été réalisée sur un corpus composé de 173 documents parlant de l'UMP qui a été découpé en 88 classes : 85% des documents ont été affectés à la bonne classe par notre outil. L'évaluation a aussi montrée que seules 5% des classes n'étaient pas compréhensibles à partir des mots-clés extraits. Cet outil est désormais utilisé par les collaborateurs de *Xeres*.

4.1.3 Autres applications

Cette sous-section survole les autres applications industrielles sur lesquelles nous avons travaillé entre 2003 et 2008. Dans la plupart des cas, nous n'avons pas pu publier d'articles sur le sujet. Pour des raisons pratiques, beaucoup de ces applications n'ont pas été évaluées de manière objective et systématique comme le réclame toute bonne procédure expérimentale.

Extraction d'entités nommées (Softissimo)

Softissimo est un éditeur de logiciels, spécialisé dans les applications linguistiques de l'informatique comme les dictionnaires ou la traduction. Dans le cadre d'une courte collaboration (en tant que consultant, juillet/septembre 2005), nous avons été chargé de mettre au point un outil d'extraction d'entités nommées et de leurs contextes. L'entreprise était notamment intéressée par les noms de personne qui, de manière générale, ne se traduisent pas. Le corpus principal d'expérimentation était un corpus de l'INA où une bonne partie des phrases étaient écrites tout en majuscule. La majuscule étant l'un des critères les plus importants pour la détection des noms de personne, il a été nécessaire de développer des heuristiques sous la forme de grammaires locales et de programmes simples. Nous avons également construit un certain nombre de lexiques spécifiques afin d'aider à l'extraction d'entités de base telles que les prénoms ou les noms de famille : fonctions, titres, noms et verbes déclencheurs. Nous avons obtenu des résultats classiques pour cette tâche : une très bonne précision, mais un rappel relativement bas.

Extraction de concepts clés (SeniorPlanet)

Ce projet a consisté à développer pour SeniorPlanet (journal en ligne) un prototype d'extraction automatique de mots-clés des articles de leur journal

électronique afin de faciliter leur travail de maintenance. Le projet, en collaboration avec E. Laporte et S. paumier, a débuté en janvier 2006 et s'est terminé par la validation du prototype par SeniorPlanet en octobre 2006. L'ensemble des mots-clés extraits doit être inclus dans un ensemble fini de mots-clés fourni par SeniorPlanet. Un mot-clé n'est pas uniquement un mot qui peut apparaître tel quel dans un texte. Il désigne aussi un concept identifiant un ensemble de mots et de séquences multi-mots. Nous avons représenté le concept associé à chaque mot-clé par un graphe d'automate fini décrivant l'ensemble des mots et des séquences multi-mots de ce concept. Chaque chemin de ce graphe comporte une sortie qui associe la séquence étiquetant le chemin à un mot-clé donné. Ces graphes ont d'abord été construit automatiquement à l'aide de la liste des mots-clés fournie par SeniorPlanet. Chaque graphe reconnaît alors la forme graphique du mot-clé associé ainsi que ses différentes formes fléchies. Les graphes ont ensuite été complétés manuellement, au moyen d'un épluchage systématique des textes fournis par SeniorPlanet et d'une introspection linguistique pour conceptualiser les différentes instances trouvées dans les textes. L'ensemble de ces graphes forme la grammaire des concepts. Le module d'extraction comportait deux phases : une phase de repérage des mots-clés potentiels du texte et une phase de sélection des meilleurs mots-clés parmi les mots-clés potentiels. La phase de repérage des mots-clés potentiels du texte consiste à appliquer la grammaire des concepts sur le texte et à extraire du résultat les séquences reconnues avec leurs mot-clés associés. La phase suivante consiste à assigner un poids de pertinence à chacun des mots-clés extraît lors de la phase précédente.

Extraction d'informations biographiques (Teragram)

Durant notre période à Teragram (2003 – 2004), un client – une entreprise spécialisée dans les ressources humaines – souhaitait extraire et structurer automatiquement des informations biographiques d'une collection de documents. Les informations biographiques demandées étaient classiques : date de naissance, études, parcours professionnel. L'idée de notre système était de relier chaque personne à chacune de ces informations au moyen de grammaires locales que nous avons construites manuellement. Afin de rassembler toutes les informations biographiques d'une même personne, il a été nécessaire d'implanter un algorithme simple de résolution de co-référence.

Affectation d'une classe flexionnelle (Teragram)

De nombreux outils de l'entreprise Teragram dépendaient de grandes ressources dictionnairiques. Il nous a été demandé de développer un petit outil affectant une classe flexionnelle à un lemme inconnu [4] afin d'aider à la mise à jour des ressources. L'algorithme est basé sur les suffixes des lemmes qui sont souvent des indices forts pour la flexion. Nous avons utilisé les dictionnaires de Teragram comme source d'apprentissage. Chaque lemme y étant associé à une classe flexionnelle, nous avons contruit une base où chaque suffixe trouvé dans les lemmes

4. Un lemme inconnu est un lemme absent des ressources lexicales.

du dictionnaire est associé à l'ensemble des classes flexionnelles correspondantes. Pour un lemme inconnu, on cherche son suffixe le plus long qui se trouve dans la base et lui associe sa ou ses classes flexionnelles. Cet algorithme simple a montré des résultats tout à fait satisfaisants.

Projet Outilex

Le projet Outilex visait à mettre à la disposition de la recherche, du développement et de l'industrie une plate-forme logicielle de traitement des langues naturelles ouverte et compatible avec l'utilisation d'XML, d'automates finis et de ressources linguistiques. En raison de son ambition internationale, Outilex a également participé aux efforts actuels de définition de normes en matière de modèles de ressources linguistiques. Le projet Outilex regroupait 10 partenaires français, dont 4 académiques [5] et 6 industriels [6]. Il était coordonné par le LIGM et financé par le ministère de l'Industrie dans le cadre du Réseau national des technologies logicielles (RNTL). Préparé sous la direction de Maurice Gross, il a été lancé en 2002 et s'est terminé en 2006. Dans le cadre de sa thèse financée par le projet, O. Blanc a été le développeur principal de la plate-forme. Lors de la dernière année du projet, nous l'avons rejoint pour aider à la finalisation. La plate-forme a été livrée en octobre 2006 et est décrite dans [Blanc et Constant, 2006, Blanc et al., 2006]. Elle n'est plus maintenue. Cependant, ses fonctionnalités principales ont été petit à petit intégrées dans Unitex.

4.2 Applications linguistiques

Notre parcours scientifique nous a aussi amené à développer des applications linguistiques, notamment pour étudier des phénomènes spécifiques (disfluences et mots composés dans des transcriptions orales) ou pour construire des ressources lexicales (extraction de traductions d'expressions multi-mots).

4.2.1 Analyse de transcriptions orales

A travers notre collaboration avec A. Dister, linguiste spécialiste de l'oral, nous nous sommes consacré au traitement des transcriptions orales. Nous avons, en particulier, travaillé sur le repérage de disfluences, phénomènes propres à l'oral. A partir des études de corpus réalisées par A. Dister, nous avons donc développé un outil nommé *Distagger* détectant ces phénomènes dans les transcriptions. A partir de cet outil et de nos travaux sur la reconnaissance des mots composés (cf. chapitre 2), nous avons conduit une étude de corpus sur le comportement des disfluences dans les mots composés. Nous avons également utilisé *Distagger* pour faire le prétraitement d'une analyse en constituants simples (cf. chapitre 2).

5. Université Paris-Est marne-la-Vallée, Université de Rouen, Université Paris 6, LORIA-INRIA.

6. Systran, Lingway, CEA, LCI, Thales Communication, Thales R & T.

Dans cette section, nous verrons d'abord les principes généraux sous-jacents à nos travaux. Nous décrirons ensuite notre approche pour repérer les disfluences et enfin nous présenterons leur comportement au sein des mots composés.

Méthodologie

Bien que les outils du TAL soient de plus en plus performants sur l'écrit, ils sont, le plus souvent, inadaptés pour l'analyse de transcriptions orales car ils doivent faire face à des problèmes spécifiques tous liés à la nature des données. Tout d'abord, pour des raisons théoriques [Blanche-Benveniste et Jeanjean, 1987], les transcriptions de l'oral ne contiennent pas de ponctuations, alors que la plupart des outils du TAL sont fondent sur une segmentation en phrases basée sur ces marques. Par ailleurs, les textes comprennent des méta-informations qui n'ont pas besoin d'analyse linguistique (noms des locuteurs, indications de chevauchement de parole, contexte énonciatif, etc.). Ensuite, les textes contiennent des particularités lexicales comme le mot *quoi* qui peut être une interjection dans un discours oral. Enfin et surtout, ces textes sont remplis de disfluences qui correspondent à des entassements brisant la linéarité syntaxique comme les amorces (e.g. *les av/*[7] *avions*), les répétitions (*les les les avions*), les auto-corrections immédiates (*la le chien*), etc. Les disfluences perturbent fortement l'analyse linguistique automatique des textes oraux comme le montrent [Adda-Decker et al., 2003, Bénard, 2005, Benzitoun, 2004, Benzitoun et al., 2004, Garside, 1995, Guénot, 2005, Nivre et Grönqvist, 2001, Oostdijk, 2003, Valli et Véronis, 1999].

Pour résoudre les problèmes que posent ces phénomènes, nous nous basons sur les résultats de [Benzitoun et al., 2004]. Ces derniers ont montré que l'annotation de corpus oraux n'était pas un problème spécifique étant donné qu'il n'existe pas de grammaire pour le langage parlé (vs. une grammaire pour le langage écrit) [Blanche-Benveniste et al., 1990]. Ceci montre qu'il n'est pas forcément nécessaire de modifier les grammaires ou règles grammaticales d'un analyseur existant pour annoter des textes oraux. Ceci tend à montrer que les outils du TAL conçus pour l'écrit sont aussi utilisables pour l'oral à condition de prétraiter les transcriptions en nettoyant les disfluences notamment. Nous avons donc développé un module de prétraitement qui normalise les transcriptions orales afin de les rendre compatibles avec des outils classiques du TAL. Le module de prétraitement est décrit dans [Constant et Dister, 2010]. L'approche adoptée se distingue donc d'approches où les outils du TAL sont directement adaptés à ces phénomènes. En particulier, [Eshkol et al., 2010] ont développé un étiqueteur morphosyntaxique probabiliste pour l'oral. Ils ont appris le modèle directement à partir d'un corpus oral annoté en catégories grammaticales où les disfluences sont conservées. [Antoine et al., 2008] ont mis au point un analyseur en constituants simples se fondant sur une cascade de transducteurs en deux étapes : une première étape avec une grammaire de chunk classique ne tenant pas compte des spécificités de l'oral, puis une deuxième étape corrigeant

7. La barre oblique indique que le mot n'a pas été complété.

la première phase en tenant compte de ces spécificités.

Les données sur lesquelles nous avons travaillé sont issues de la banque de données textuelles orales Valibel [8]. Les conventions utilisées dans les transcriptions sont explicites [Dister et al., 2009b] et convergent largement avec celles adoptées dans d'autres projets (cf. Corpus de référence du français parlé de l'équipe DELIC, les données du projet Rhapsodie, pour ne citer qu'eux).

Repérage des disfluences

Dans cette partie, nous précisons l'algorithme itératif de repérage des disfluences que nous avons mis en oeuvre. Les transcriptions que nous traitons ne contiennent aucune information préalable sur les disfluences, sauf un indice sur les mots amorcés (barre oblique indiquant une incomplétude). Il existe de nombreux travaux sur le repérage de telles séquences, que ce soit directement dans des données orales parlées ou dans des transcriptions manuelles ou automatiques : par exemple, [Lendvai et al., 2003, Snover et al., 2004, Liu et al., 2006]. Une bonne partie de ces travaux est basée sur des modèles statistiques utilisant différents traits du corpus (valeurs lexicales, prosodie, catégories grammaticales, etc.). Ne possédant pas de corpus de grande ampleur déjà annoté en séquence disfluente qui pourrait nous servir de corpus d'apprentissage, il est impossible d'utiliser de telles approches. Nous avons donc créé un système basé sur des règles symboliques simples avec peu de ressources n'impliquant aucun processus automatique linguistique préalable (ex. étiquetage grammatical). Il se fonde en particulier sur une étude linguistique systématique des disfluences dans un corpus de 440.000 mots transcrits de l'oral [Dister, 2007]. Il implémente un algorithme itératif reconnaissant des disfluences simples et permettant de former des groupes de disfluences simples entrelacées. Notons que [Bouraoui et Vigouroux, 2009] proposent un outil pour détecter les disfluences dans un corpus spécialisé [9] se basant sur une approche symbolique et sur le schéma d'annotation des disfluences de [Bear et al., 1993]. Leur outil utilise notamment une grammaire hors contexte et des classes lexicales sémantiques. Ils considèrent, en particulier, que deux mots appartenant à la même classe lexicale et se trouvant dans un voisinage proche dans le texte sont des membres potentiels de disfluences. Dans notre cadre, le corpus appartient à la langue générale. Il est donc difficile de former de telles classes sémantiques (du moins, à faible coût).

L'idée de notre système est de repérer et de classer les séquences disfluentes, pour ne conserver dans le texte que la séquence réparée afin de les soumettre à un analyseur linguistique de données plus standard. La classe et les positions initiales des séquences disfluentes sont sauvegardées afin de pouvoir les réutiliser et les réinsérer dans le texte une fois que ce dernier a été analysé. Ainsi, après passage dans notre système, l'exemple **ilePA2**

8. Ces corpus forment aujourd'hui le plus grand corpus informatisé de données textuelles orales en francophonie : une banque de données de près de 4 millions de mots. Voir http ://www.uclouvain.be/valibel-corpus.html

9. Enregistrements de dialogues oraux spontanés entre des contrôleurs aériens en formation et des "pseudos-pilotes"

ilePA2 *or une trémie euh grammaticalement c'est une chose qui s'en/ qui s'enfonce plutôt dans la terre*

devient

ilePA2 *or une trémie grammaticalement c'est une chose qui s'enfonce plutôt dans la terre*

Les positions initiales de *euh* (type euh) et *qui s'en/* (type amorce) sont sauvegardées afin de les réintégrer par la suite.

L'article [Constant et Dister, 2012] décrit, entre autres, l'algorithme itératif et l'évaluation de *Distagger*. Une disfluence simple est relativement facile à repérer à l'aide de patrons simples. Cependant, les combinaisons de plusieurs disfluences entrelacées les unes dans les autres complexifient la tâche. Une solution est d'écrire différents patrons correspondant à chacune des combinaisons possibles, avec l'inconvénient de devoir écrire de multiples règles. Une autre, celle que nous avons choisie, consiste à n'appliquer qu'un seul patron de manière itérative jusqu'à obtenir un point fixe. Le patron appliqué est composé de trois parties : une séquence w de mots, une séquence d'insertions I (pauses silencieuses, mots d'éditions e.g. *ben*, ...) et une séquence c de mots (potentiellement vide) qui correspond à une correction de w. Chaque mot de c est une correction du mot correspondant de w (quand il en a un) : soit le mot lui-même, soit un mot appartenant à sa classe d'équivalence [10] définie par l'utilisateur (ex. {*le*, *la*, *les*}), soit un mot dont w est le préfixe dans le cas d'une amorce (ex. *j/* pour *je*). Lors de l'application de ce patron à une position donnée du texte, s'il correspond, la séquence wI est supprimée pour ne conserver que la correction c. L'algorithme consiste à faire plusieurs applications glissantes du patron de reconnaissance sur le texte jusqu'à ce que ce dernier ne soit plus modifié. Une fois toutes les séquences disfluentes repérées, celles-ci sont typées à l'aide de règles simples utilisant le type de correction détectée. L'ensemble de la procédure est implanté dans l'outil *Distagger* [Constant et Dister, 2010]. L'évaluation de notre outil de détection des disfluences a consisté à l'appliquer sur deux transcriptions extraites de nos données, qui ont la caractéristique d'avoir des locuteurs dont le taux de disfluences est supérieur aux autres [Dister, 2007]. Le résultat de cette application a été confronté à l'annotation de référence validée manuellement. Ce corpus de référence comprend au total 1297 tours de parole, 22476 mots graphiques et 1280 disfluences. Au total, notre système atteint une f-mesure de 95,5%. L'outil *Distagger* est librement disponible sous licence LGPL. Il a été utilisé par I. Eshkol pour des travaux pratiques à l'Université d'Orléans. Quelques essais ont également été réalisés sur le corpus du projet ANR Raphsodie (sans lendemain...).

10. Cette classe ressemble aux classes lexicales sémantiques proposées par [Bouraoui et Vigouroux, 2009]. Nous pourions donc réutiliser leurs classes telles quelles pour traiter leur corpus spécialisé.

Etude des disfluences dans les mots composés

A partir des outils développés, nous avons réalisé une étude sur les disfluences dans les mots composés. Les disfluences ont la particularité de briser la linéarité syntaxique de l'énoncé dans lequel elles apparaissent. Elles constituent une interruption (souvent momentanée, parfois définitive) dans le déroulement de l'énoncé. Les expressions polylexicales telles que les mots composés forment des unités syntaxiques et sémantiques. Du fait de cette double propriété, l'énonciation de telles expressions dans un discours oral nous paraît moins propice à l'apparition de disfluences qu'une séquence libre de mots. C'est l'hypothèse que nous avons vérifié dans [Constant et Dister, 2012].

Afin de vérifier notre hypothèse, nous avons utilisé la procédure suivante. Nous sommes partis d'un corpus de transcriptions orales formé de près de 500 000 mots graphiques [Dister, 2007]. Nous avons repéré les disfluences à l'aide de l'outil *Distagger* [Constant et Dister, 2010] présenté ci-dessus. Les disfluences ont alors été supprimées du texte afin de rendre à ce dernier sa linéarité syntaxique. Nous avons ensuite balisé les mots composés au moyen du segmenteur-étiqueteur *lgtagger* basé sur le modèle probabiliste des champs aléatoires markoviens et sur un lexique morphosyntaxique à large couverture [Constant et Sigogne, 2011]. Le segmenteur-étiqueteur a été appris à partir de deux corpus annotés en parties-du-discours où les mots composés sont marqués : le French Treebank [Abeillé et al., 2003] et le corpus oral utilisé dans [Eshkol et al., 2010]. Les disfluences repérées initialement ont alors été réinsérées dans le texte balisé afin de procéder à divers calculs statistiques nous permettant de vérifier notre hypothèse de départ.

Notre corpus de travail comprend exactement 478 084 unités dont 22 205 unités disfluentes : une unité est soit une séquence disfluente soit un mot graphique. On constate ainsi que 4,6% des unités du corpus sont disfluentes. Par ailleurs, le corpus comprend 15 350 mots composés repérés automatiquement, ce qui correspond à 38 400 unités. Par contre, seules 2,7% des unités des mots composés sont des séquences disfluentes (soit 40% de moins que la distribution sur le texte entier). Ce résultat confirme notre hypothèse de départ de manière claire. Par ailleurs, nous avons examiné les positions des disfluences dans les mots composés. Plus précisément, nous avons regardé deux positions particulières. Les disfluences qui se trouvent en position initiale du mot composé (ex. *ca/ carte bancaire* ; les disfluences en position interne (*carte euh bancaire*). Nous observons qu'elles ont tendance à se trouver en position initiale (89 % d'entre elles contre 11% en position interne). Ceci revient à dire qu'une fois le mot composé bien amorcé, son énonciation tend à être linéaire. On observe que les prépositions et la locution *il y a* sont particulièrement sujettes aux disfluences. Ceci peut peut-être s'expliquer par le fait qu'elles jouent le plus souvent le rôle d'introducteur de chunk nominal, et se présentent donc dans l'énoncé à un moment où le locuteur est à la recherche de la dénomination [Blanche-Benveniste, 1985]. Par contre, les adverbes, les noms et les conjonctions ont tendance à être moins propice à une disfluence.

4.2.2 Extraction de lexiques bilingues d'expressions multi-mots

Notre carrière d'enseignant nous a amené à délivrer un cours sur l'alignement dans des corpus parallèles et, pour cette occasion, à découvrir les modèles probabilistes associés à cette tâche. Par jeu, nous nous sommes intéressé à l'alignement bilingue d'expressions multi-mots. Etant donnée une expression multi-mots dans une phrase dans une langue source, le but est d'aligner cette séquence avec sa traduction dans la phrase correspondante en langue cible. Une telle tâche permet notamment d'aider à construire des lexiques bilingues d'expressions multi-mots qui posent de sérieux problèmes à la traduction automatique. De plus en plus d'expériences ont été réalisées sur ce sujet. Certaines utilisent des méthodes statistiques (par exemple, [Smadja et al., 1996, Caseli et al., 2009, Bai et al., 2009]). D'autres exploitent des méthodes plus linguistiques comme [Seretan et Wehrli, 2007] qui se basent sur les résultats d'un analyseur syntaxique. Enfin, il existe des méthodes hybrides [Lü et Zhou, 2004, Deléger et al., 2009, Morin et Daille, 2010]. La plupart du temps, elles traitent des collocations ou des termes et utilisent des corpus parallèles multilingues. De nombreux travaux, notamment pour la traduction de termes, utilisent des approches compositionnelles à partir de lexiques bilingues comme le montrent [Morin et Daille, 2010]. Ces derniers ont aussi montré qu'appliquer des transformations syntaxiques et morphologiques améliorent les résultats. D'autres, comme [Caseli et al., 2009, Bai et al., 2009], se basent uniquement sur des corpus parallèles. C'est dans ce cadre que nous nous plaçons. Il existe deux approches statistiques principales d'alignement de séquences de mots : (a) soit au moyen de modèles probabilistes de traductions de type IBM ; (b) soit au moyen de mesures de corrélation se basant sur le contexte lexical. De telles études sont des cas particuliers du *transpotting* qui cherche à aligner n'importe quelle séquence de mots (compositionnelle ou non compositionnelle) d'une langue source vers une langue cible, comme c'est le cas dans le concordancier bilingue *TransSearch* [Bourdaillet et al., 2010].

Dans nos travaux, nous avons traité les mots composés et les constructions à verbe support. Nous confrontons les méthodes utilisées pour les collocations à ces deux types d'expressions multi-mots. Nous avons, entre autres, utilisé l'outil Giza++ [Och et Ney, 2003] qui permet d'aligner mot à mot les phrases correspondantes dans un corpus parallèle bilingue. Il se base sur les modèles de traductions IBM de 1 à 5 [Brown et al., 1993]. Notre corpus de travail est Europarl [Koehn, 2005]. Ce corpus parallèle librement disponible sur Internet provient des actes du Parlement Européen et inclut des versions en 11 langues européennes. Chaque langue comprend environ 1 million de phrases, qui contiennent de l'ordre de 28 millions de mots. Europarl est en général considéré comme un corpus spécialisé pour deux raisons : la structuration du discours est très formatée ; le corpus fourmille de termes spécialisés. Malgré cela, il est intéressant pour notre étude car les phrases utilisées ont des structurations syntaxiques très variées et il existe un grand nombre de mots du langage général. Nous nous sommes limités à deux paires de langues : (1) français-anglais pour nos expériences sur les mots composés et (2) italien-anglais pour nos expériences

sur les constructions à verbe support.

Extraction de traductions de mots composés

Dans un premier temps, nous avons traité les mots composés, séquences de mots contigus non-compositionnelles, qui sont présentes dans le dictionnaire DELACF [Courtois et al., 1997]. Cette étude expérimentale est décrite dans [Constant et al., 2010]. Nous y confrontons les méthodes utilisées pour les collocations aux mots composés. Alors que les collocations ont tendance à mettre en relation deux mots pleins (ex. verbe-nom pour les collocations verbe-objet, ex : *prendre l'apéritif* ; nom-adjectif pour les collocations nominales : *pain perdu*), certains types de mots composés comme les prépositions ne possèdent souvent qu'un seul mot plein entouré de mots grammaticaux (*au sein de*), ce qui les rend plus difficile à repérer et traduire que les collocations traditionnelles. Les mots composés que nous traitons appartiennent à quatre catégories : les noms, les adverbes, les conjonctions et les prépositions. Ceci sont reconnus à l'aide de l'analyseur en constituants simples décrit dans [Blanc et al., 2007] qui inclut la reconnaissance des mots composés en exploitant des ressources lexicales (cf. chapitre 2). Pour notre étude, nous nous sommes limités au dictionnaire DELACF [Courtois et al., 1997]. La proportion de mots composés français mal identifiés automatiquement est relativement modérée : environ 4%. Nous n'avons travaillé qu'avec les mots composés correctement identifiés. Pour notre tâche d'alignement, nous nous basons sur les études réalisées sur l'extraction statistique des traductions de collocations. Celles-ci se fondent sur les modèles probabilistes IBM d'alignement comme dans [Caseli et al., 2009] ou sur des mesures de corrélation [Bai et al., 2009].

Une méthode basique (méthode BASIC) consiste à aligner mot à mot les phrases du corpus parallèle, en considérant les mots composés français comme des mots simples (ex. *au sein de* → *au_sein_de*). Pour cela, il suffit d'utiliser un aligneur mot à mot du type Giza++ sur un corpus parallèle où les mots composés auront été identifiés au préalable. Théoriquement, un mot composé étant une unité élémentaire, c'est la méthode la plus intuitive. Or, du fait de la distribution des mots composés dans les corpus, l'apprentissage s'avère difficile. En effet, un mot composé donné apparaît peu souvent. Donc les méthodes purement statistiques ont du mal à apprendre leurs comportements. En pratique, une partie des mots composés se traduisent à partir de leurs composants simples. Étant donné cela, une méthode d'identification des traductions des mots composés est d'aligner les mots graphiques des phrases parallèles (méthode GIZA). On considérera alors que la traduction d'un mot composé sera l'union des alignements de ses composants simples.

De nombreuses méthodes d'extraction de traductions des collocations utilisent des mesures d'associativité. Ces dernières calculent le degré de corrélation d'un mot ou un groupe de mots en langue source avec un mot ou un groupe de mots en langue cible. Parmi les mesures de corrélation les plus populaires, la mesure de Dice est très efficace. [Bai et al., 2009] estiment néanmoins qu'elle a plusieurs défauts. En particulier, les mots composés ont parfois un lien de collocation

fort avec leur contexte. Il est donc nécessaire de tenir compte de celui-ci pour calculer le degré de corrélation entre un mot en langue cible et l'expression en langue source. Ainsi, ils ont mis au point le principe de fréquence de corrélation normalisée qui tient compte du contexte dans lequel le mot composé est plongé. Nous avons donc implanté leurs formules décrites dans [Bai et al., 2009] (méthode CORR).

Toutes ces stratégies ont été évaluées à l'aide d'un corpus que T. Nakamura, S. Voyatzi et A. Bittar ont annoté. Ce corpus comporte 1002 phrases contenant 26422 mots français et 25082 mots anglais. Il inclut un total de 998 mots composés, dont 532 différents. Ce corpus de référence a été construit semi-automatiquement de la manière suivante : nous avons d'abord appliqué la méthode GIZA de repérage des traductions dans les différentes phrases. Les annoteurs ont ensuite vérifié et post-édité manuellement (en cas d'erreur) les occurrences des mots composés français identifiés et pour chacun d'eux, la traduction repérée. Ils ont couvert la même part du corpus et les annotations obtenues concordaient à environ 85%. La principale source de désaccord était la traduction des prépositions composées qui sont souvent traduites de manière détournée, par exemple via des reformulations libres des phrases traduites, ce qui provoque des difficultés dans l'annotation. Les erreurs d'inattention forment l'autre part des erreurs. Les désaccords ont été systématiquement analysés et ont été résolus après discussion.

Pour l'ensemble des mots composés traduits, la méthode GIZA est la meilleure avec près de 68% de précision dans les traductions. Comme prévu, la méthode BASIC est beaucoup plus faible avec une précision de 56% environ. La méthode CORR est intermédiaire et a une précision autour de 63%. Les résultats globaux sont donc relativement moyens, ce qui s'explique par la difficulté de la tâche. Si l'on regarde maintenant les résultats par catégorie grammaticale, on s'aperçoit que les deux meilleures approches (GIZA et CORR) sont complémentaires (cf. figure 4.1). En effet, pour les adverbes et les noms, la méthode CORR dépasse GIZA. Les performances de CORR s'écroulent pour les prépositions et les conjonctions. Ce n'est d'ailleurs pas une surprise car il est bien connu que les mesures associatives capturent mal le comportement des mots grammaticaux. Du coup, les méthodes utilisant ces mesures sont grandement désavantagées dans les expressions contenant une majorité de tels mots, comme c'est le cas pour les conjonctions et les prépositions. Une méthode comme GIZA limite les dégâts grâce à l'utilisation du contexte de la phrase entière. Cette analyse des résultats par catégorie grammaticale est extrêmement intéressante : elle montre que l'on pourrait améliorer les résultats si l'on utilisait la méthode GIZA pour les prépositions et les conjonctions et la méthode ASSOC pour les noms et les adverbes.

Une future expérience pourrait comparer les différences méthodes selon le degré de figement des mots composés. Par ailleurs, il existe certaines erreurs fréquentes : les traductions proposées sont discontinues au lieu d'être continues. Nous avons résolu, en partie ces problèmes, par des heuristiques post-traitement. Pour notre méthode GIZA, nous pourrions utiliser l'approche probabiliste de [Simard, 2003] implanté dans *TransSearch* [Bourdaillet et al., 2010].

Catégorie	Répartition	CORR	GIZA
Adverbe	8%	65	63
Conjonction	5%	45	63
Nom	65%	74	73
Préposition	22%	32	56

FIGURE 4.1 – Résultats en terme de précision par catégorie grammaticale

Des méthodes incluant des processus plus linguistiques comme dans [Seretan et Wehrli, 2007, Morin et Daille, 2010] semblent également prometteuses.

Extraction de traductions d'expressions à verbe support

Nous avons ensuite étendu nos travaux aux expressions à verbe support (CVS) en utilisant une stratégie d'alignement proche de celle réalisée par [Samardzic et Merlo, 2010]. Dans [Constant et Guglielmo, 2010], nous expérimentons uniquement la méthode GIZA vue dans la section précédente que nous adaptons aux CVS des noms predicatifs de l'italien vers l'anglais. Les noms prédicatifs (Npred) que nous avons étudié entrent dans la construction de base suivante :

N0 Vsup (E+Det) Npred (E + Prep N1)
= : Max ha appetito (Max a de l'appétit)

Le nom prédicatif possède un verbe support au sens vide (Vsup). Un Vsup peut aussi être remplacé par une variante qui porte une valeur sémantique [Gross, 1981]. Par exemple, dans *Max perde l'appetito* (Max perd l'appétit), *perdere* est une variante négative du verbe *avere* (avoir). Ainsi, un même nom prédicatif peut posséder de nombreuses variantes de verbe support qui réalisent une classe d'équivalence :

La Renault (stringe+fa+ha+ mantiene+ firma) accordi con la Mercedes
(Renault (fait+a+maintient+signe) un accord avec Mercedes)

D. Gugliemo a rassemblé un ensemble de 300 entrées nominales en italien sous forme d'un lexique tabulaire. Pour chacune d'elles, elle a codée les diférentes valeurs lexicales associées : verbes supports, variantes de verbes supports, déterminant, préposition du complément N1 (s'il est présent). L'identification automatique des CVS en italien est basée sur un ensemble de grammaires locales générées automatiquement à partir de nos ressources lexicales. Nous utilisons le principe des graphes paramétrés du logiciel Unitex [Paumier, 2003] pour la génération. Un graphe paramétré décrit un patron syntaxique générique des CVS, paramétré par les propriétés lexico-syntaxiques de notre lexique. Pour chaque entrée de la table, le programme fait une copie du graphe et résoud les paramètres en fonction du codage de l'entrée dans le lexique. Ainsi, à par-

tir du graphe paramétré, nous générons un graphe pour chaque entrée lexicale, qui reconnaît les différentes structures dans lesquelles elle entre. L'ensemble des graphes est alors appliqué à notre texte italien pour reconnaître les différentes CVS décrites.

Une fois les CVS repérées, il est nécessaire de les aligner avec leur traduction dans le texte correspondant en anglais. Nous utilisons donc la méthode GIZA décrite dans la section 4.2.2. Pour rappel, l'idée est d'aligner mot à mot les différents phrases à l'aide de Giza++. On considére alors que la traduction d'une CVS repérée dans le texte italien sera l'union des alignements de ses mots simples en anglais.

Au total, 450 occurrences de CVS (décrits dans la table) on été trouvés et balisés dans la version italienne de notre corpus de travail. Pour chacune d'elles, nous avons appliqué notre procédure automatique d'alignement. Nous avons alors manuellement comparer la traduction générée pour notre outil et la traduction de référence. Notre procédure d'évaluation n'est pas binaire et comporte une certaine subjectivité car il existe différents degrés de correction. Nous avons considéré 4 types de traductions prédites :

1. Correcte : quand la traduction est exacte (*porre una domanda* → *ask the question*)
2. Partiellement correcte : quand la traduction trouvée est seulement une partie de la traduction de référence (*dato risposta* → *response* instead of *provide a response*)
3. Incorrecte : quand la traduction trouvéé est fausse (*prenda l' iniziativa* → *it is important lead* instead of *take the lead*) ;
4. Pas de sortie : quand aucune traduction n'est identifiée, bien qu' elle existe dans le corpus

On observe que 55% des traductions sont correctes, 22% sont partiellement correctes, 17% sont incorrectes et 6% n'ont pas de sortie de manière erronnée. Ces résultats montrent que la procédure est largement perfectible. On pourrait notamment utiliser une méthode similaire à [Seretan et Wehrli, 2007] qui exploitent les résultats d'un analyseur syntaxique pour des collocations de type V-N. Cependant, nous sommes mesurés car de nombreuses erreurs proviennent d'une des trois causes suivantes : (1) les erreurs du corpus ; (2) des ambiguités lexicales impliquant des stratégies de résolution d'ordre sémantique ; (3) le choix (par les interprètes) de traduire les CVS italiennes par des paraphrases.

Chapitre 5

Ressources linguistiques

Dans les chapitres précédents, nous avons vu l'intérêt indéniable des ressources lexicales et des corpus annotés pour la prise en compte des expressions-multi-mots dans les analyseurs ou les applications du TAL. Cependant, il existe un manque criant de ressources qui explique en partie le nombre limité de travaux d'intégration de la reconnaissance des unités polylexicales. Pour palier très partiellement ce manque, nous avons donc mis notre pierre à l'édifice des ressources lexicales et des corpus annotés. Dans ce chapître, nous retraçons rapidement nos premiers pas dans le domaine autour justement des ressources lexicales. Puis, nous décrirons quelques tentatives plus récentes de construction de corpus annotés. Nous présenterons également quelques outils que nous avons développés pour compiler et gérer les ressources. Ce chapitre est volontairement court car cet axe de recherche ne se trouve plus au coeur de nos problématiques principales actuelles.

5.1 Construction

5.1.1 Ressources lexicales

Notre thèse [Constant, 2003] a été en partie consacrée à la description de phénomènes linguistiques particuliers liés aux entités nommées : les expressions de mesure et les groupes prépositionnels géographiques. Pour l'étude des expressions de mesure, nous sommes parti des travaux décrits dans [Giry-Schneider, 1991]. Pour les groupes prépositionnels géographiques, nous sommes parti des travaux sur les constructions *être Prep X* de [Danlos, 1980, Gross, 1996b], des études sur les constructions verbales locatives de [Guillet et Leclère, 1992], ainsi que des travaux sur les toponymes de [Piton et al., 1999].

Nous nous sommes appuyé sur la méthodologie du lexique-grammaire [Gross, 1975] en ramenant chaque phénomène à des phrases élémentaires. Pour les mesures, nous nous sommes ramené à des phrases du type *L'immeuble a une hauteur de 260 m* ou *Paris est à une distance de 200 km de Lille* qui impliquent une valeur mesurant la propriété d'une entité (*hauteur*) ou la propriété reliant

deux entités (*distance*). Pour les groupes prépositionels, nous avons travaillé sur des phrases comme *Max est à l'île de la Réunion* qui impliquent principalement une préposition locative, un classifieur de lieu géographique et un nom propre désignant le lieu. Nous avons d'abord étudié en détail les composants élémentaires de ces phrases. Pour les mesures, nous avons décrit sous la forme de grammaires locales les déterminants numériques, les unités de mesure et leur combinaison (*dix mètres* ; *10 à 15 milles marins*, etc.). Pour les groupes prépositionnels, nous avons systématiquement étudié les noms propres géographiques et codé leurs contraintes internes sous forme de lexique tabulaire. Par exemple,

(18) la mer (Méditerranée + Noire) = La (Méditerranée + *Noire)

(19) la mer (*de Glace + du Nord) est une mer

(20) La Manche est une mer/* la mer (E+ de + de la) Manche

(21) l'île (*E +de la) Réunion

(22) l'île (E + *de) Maurice)

Nous avons aussi analysé les prépositions locatives composées comme *à 30 km à l'ouest de* ou *au milieu de*.

Ensuite, en appliquant systématiquement des transformations syntaxiques sur nos phrases, nous avons observé différentes contraintes syntaxiques et lexicales que nous avons aussi codées dans des lexiques tabulaires sur le même principe que les tables du lexique grammaire. Par exemple, pour les expressions de mesure :

(23) La corde a une longueur de 10 m = La corde fait 10 m de long

(24) La piscine a une profondeur de 2 m = La piscine fait 2 m de (*profond+fond)

Pour les groupes prépositionnels, on observe entre autres les contraintes suivantes :

(25) Luc est (en + *E) mer (du Nord + Noire+ Méditerranée)

(26) Léa est (*en + E) rue (de la Paix + Monge)

(27) Max est (*à la + en) (Crète + Corse)

(28) Max est (à la + *en) Réunion

Nous établissons ci-dessous une synthèse de nos publications sur ces deux sujets. Nos publications ne couvrent qu'une partie de nos études linguistiques de thèse. Les articles suivants sont liés aux expressions de mesure. [Constant, 2000] est notre premier article sur le sujet. Il décrit des grammaires locales de diverses expressions numériques, incluant les expressions de mesure. Dans [Constant, 2002a], nous avons synthétisé plusieurs méthodes pour construire des ressources sur les expressions de mesure dans le cadre du lexique-grammaire. Nous montrons quelques exemples d'applications comme l'interprétation sémantique des mesures. Dans [Nakamura et Constant, 2001], nous étudions les expressions de

pourcentage, que nous intégrerons dans les expressions de mesure dans notre thèse. Dans [Constant, 2009], nous avons présenté un processus de construction et d'évaluation de grammaires locales spécifiques à la valeur des mesures (ex. *25 centimètres*, *entre 2 et 5 kg*). Nous avons montré que ces séquences a priori simples possèdent des contraintes complexes. Les publications suivantes sont liées aux groupes prépositionnels géographiques. [Constant, 2002b] contient une brève description linguistique des groupes prépositionnels géographiques. Nous montrons ensuite un processus d'application et d'évaluation des grammaires construites semi-automatiquement à partir des lexiques tabulaires dérivés de l'étude linguistique. Dans [Constant, 2010], nous réalisons une description plus approfondie des groupes prépositionnels géographiques.

Dans nos futurs travaux de recherche, nous ne souhaitons pas continuer sur ces sujets. Peut-être reviendrons-nous une dernière fois sur les expressions de mesure le temps d'un article de revue.

5.1.2 Corpus annoté

Annotation de *faire* dans un corpus oral

En collaboration avec A. Dister et T. Nakamura, nous avons étudié les constructions en *faire* dans des transcriptions orales. Ce verbe entre souvent dans des expressions multi-mots comme les constructions à verbe support et les expressions figées. En s'appuyant sur une classification basée sur des critères formels et en exploitant des informations présentes dans les tables du lexique-grammaire, nous avons annoté quelques milliers de telles constructions dans un sous-corpus du corpus Valibel [Dister et al., 2009b]. Notre article [Constant et al., 2013] décrit cette étude linguistique, la procédure d'annotation mise en place et les résultats obtenus. L'étude linguistique nous a amenés à établir une typologie des différents emplois du verbe *faire* trouvés dans le corpus. En partant de la tripartition classique du lexique-grammaire (constructions libres, à verbe support et figés), de l'étude de [Giry-Schneider, 1987] et des différentes observations sur les données, nous avons réparti les occurrences de *faire* en 7 classes : (1) les emplois causatifs (cf. exemple 29) ; (2) les emplois "passe-partout" (cf. exemple 30) ; (3) les emplois "support" (cf. exemple 31) ; (4) les emplois figés (cf. exemple 32) ; (5) les emplois semi-figés (cf. exemple 33) ; (6) les emplois pro-verbe (cf. exemple 34) ; (7) les autres emplois. Les emplois "passe-partout" correspondent à des occurrences de *faire* qui peuvent être remplacées par un verbe plein. Les emplois semi-figés rentrent dans des constructions syntaxiques figées, mais admettent un complément avec une certaine variation lexicale (le plus souvent appartenant à une classe sémantique claire). De nombreuses observations faites dans cet article peuvent aussi se retrouver dans [Danlos, 1994].

(29) je crois que je vais la faire agrandir

(30) il ne faut que deux semaines pour (faire+fabriquer) un châssis

(31) on va faire des achats

(32) recommencer tout / et euh / faire table rase du wallon

Type	Nombre d'occurrences	Pourcentage (%)
Causatif	476	16
Passe-partout	294	10
Support	1024	33
Figé	287	9
Semi-figé	243	8
Pro-verbe	567	19
Autres	8	0
Non exploitable	166	5
Total	3045	100

TABLE 5.1 – Répartition des emplois de *faire*

(33) Ça fait DUREE que P = : Ça fait (15 ans+longtemps+3 minutes+...) que j'attends

(34) alors ce qu'on essaie de faire / c'est au début du camp ils apportent leurs enfants ils disent au revoir aux enfants et puis / ils partent

Le codage systématique des 3 045 occurrences du verbe *faire* (et ses formes fléchies) dans notre corpus a conduit à la répartition donnée dans la table 5.1. On observe que 5% des occurrences n'ont pu être codées. Les raisons en sont diverses, mais tiennent en général au fait que le discours est interrompu, ce qui ne permet pas l'identification du complément de *faire*. On a également dans cette catégorie les emplois qu'il nous faut encore classer, et que nous laissons actuellement de côté. On notera que les codeurs ont exprimé un doute sur 10% des codages réalisés (ce qui montre en partie la complexité de la tâche). Les constructions les plus fréquentes en corpus sont celles à verbe support, avec un tiers des occurrences. Elles sont suivies par les emplois de faire comme pro-verbe (19% des cas), puis par les constructions causatives (16%). Sur l'ensemble des données, 4% des occurrences relèvent d'un emploi pronominal de *faire*. En ce qui concerne les emplois causatifs, *faire* est suivi d'une infinitive dans 55% des cas et d'une complétive dans 17% des cas. Environ 12% des emplois causatifs sont figés. Parmi les occurrences de *faire* comme verbe support, on note que 75% d'entre elles sont répertoriées dans la version 3.3 du lexique-grammaire [Tolone, 2011] et que 29% sont des nominalisations de verbes pleins à la [Giry-Schneider, 1978]. En cumulant les emplois figés des catégories figées et causatifs, on recense 11% d'occurrences figées dans le corpus, dont environ deux tiers sont répertoriées dans le lexique-grammaire des expressions figées.

Annotation sémantique et multilingue de verbes

Le travail sur la levée d'ambiguité sémantique de M. Rakho dont nous co-encadrons la thèse l'a amenée à construire un corpus annoté pour apprendre et évaluer ses modèles [Rakho et al., 2012]. Elle a utilisé une sous-partie d'Europarl [Koehn, 2005] Français-Anglais incluant un ensemble prédéfini de 20 verbes français très polysémiques. Comme le suggèrent [Ide et Wilks, 2006], elle a proposé de combiner des informations multilingues provenant de traductions et des informations monolingues provenant d'un lexique syntaxique, dans le but de déterminer les sens des mots polysémiques. Elle a annoté les verbes selon leur identifiant dans les tables du lexique-grammaire [Gross, 1975]. Cet identifiant correspond à un emploi lexical (i.e. un sens). Cet emploi peut être libre ou figé. Elle a également annoté de manière semi-automatique les traductions des verbes trouvées dans le corpus parallèle. Nous donnons quelques statistiques sur le contenu de ce corpus dans la table 5.2. Dans cette table, # *occ.* indique le nombre d'occurrences de chacun des verbes, # *sens* correspond au nombre d'emplois associés au verbe dans le lexique-grammaire, # *trad.* indique le nombre de traductions trouvées dans le corpus pour le verbe. A partir de ce corpus, M. Rakho a pu notamment associé pour chaque sens en français l'ensemble de ses traductions en anglais.

Verbe	# occ.	# sens	# trad.
arrêter	2033	12	150
comprendre	8240	8	183
conclure	3488	5	79
conduire	2114	10	96
connaître	5786	14	158
couvrir	2183	16	85
entrer	2325	6	107
exercer	1851	4	86
importer	2778	5	71
ouvrir	2656	17	127
parvenir	7469	3	152
porter	3301	20	219
poursuivre	5354	5	154
rendre	6731	14	177
tirer	2163	19	102
venir	7369	12	120
Total	3837	11	129

TABLE 5.2 – Statistiques sur le corpus annoté, tirées de [Rakho et al., 2012]

5.2 Outillage

Nos travaux sur les ressources lexicales nous a conduit à réaliser un certain nombre d'outils facilitant leur gestion et leur exploitation pour le TAL. Dans cette synthèse, nous montrons trois exemples : (1) la construction d'une bibliothèque décentralisée de grammaires locales ; (2) la transformation de lexiques tabulaires en automates ; (3) la transformation de lexiques tabulaires en un format XML explicite pour le TAL.

5.2.1 Une bibliothèque décentralisée de grammaires locales

Dans le but de faciliter la collaboration entre auteurs de grammaires locales, nous avons mis au point un réseau de grammaires dans une infrastructure décentralisée. Ce réseau est décrit dans [Constant et Watrin, 2008]. Ce système, nommé Graal, ouvert à tous, avait pour ambition de (1) proposer un support simple à des chercheurs isolés pour diffuser leurs grammaires locales, (2) faire office, à terme, d'état-de-l'art dans le domaine des grammaires locales, (3) permettre une utilisation intensive des grammaires locales dans des applications du TAL au moyen d'outils d'importation. Cette bibliothèque est accessible en-ligne de manière transparente au moyen de l'applet GraalWeb [1]. Elle est limitée aux grammaires au format Unitex [Paumier, 2003].

Le système Graal comporte un ensemble de serveurs HTTP de grammaires locales. Ces serveurs jouent l'unique rôle de "dépôt" et sont gérés indépendamment les uns des autres par leurs propriétaires respectifs. Un dépôt est défini par une URL de base et un propriétaire. Par exemple, il peut être situé sur le propre site web d'un auteur de grammaires. Par ailleurs, les grammaires d'un dépôt peuvent faire appel à des grammaires d'autres dépôts, ce qui créé naturellement un réseau de grammaires locales. Un utilisateur ou une application souhaitant avoir accès à leur contenu passent par un serveur d'accès. Ce serveur comporte un index à partir duquel toutes les requêtes sont traitées. L'architecture décentralisée est ainsi transparente pour l'utilisateur. L'index détient des informations précises sur la bibliothèque, notamment sur le contenu linguistique des différents dépôts ce qui permet aux utilisateurs de définir des requêtes spécifiques sur le contenu, lexical notamment. L'applet java GraalWeb donne une vue en-ligne de Graal au moyen d'un moteur de recherche et d'un explorateur. Elle permet aussi de télécharger les paquetages de grammaires disponibles dans la bibliothèque. Le moteur de recherche utilise des techniques classiques du domaine de la recherche d'informations. L'explorateur permet d'avoir une vue d'ensemble de la bibliothèque (l'ensemble des grammaires, les dépendances entre les grammaires) et une vue détaillée du contenu au moyen d'un visualisateur avancé d'automates.

Contrairement à nos attentes, la mayonnaise n'a pas vraiment pris (cinq dépôts seulement). Il existe plusieurs raisons à cela. Tout d'abord, nous n'avons certainement pas assez communiqué sur le sujet. La procédure pour créer un

1. http ://igm.univ-mlv.fr/m̃constan/library/

dépôt était peut-être trop lourde. Ensuite, les chercheurs étaient plus enclins à télécharger les grammaires existantes qu'à partager leurs propres grammaires. Enfin et surtout, GraalWeb n'est pas localisé dans la plateforme Unitex. Notons que, depuis peu, dans le projet GramLab [2] proposant une extension d'Unitex, il est prévu de mettre en place un système de dépôt (indépendant de Graal) afin de favoriser le partage des grammaires locales.

5.2.2 Transformer des lexiques tabulaires en automates

Nous nous sommes penché sur la transformation (ou compilation) en automates de contraintes linguistiques codées dans des matrices (lexiques tabulaires), dans le but de permettre la reconnaissance, dans des textes, des phénomènes codés dans ces lexiques. En particulier, en collaboration avec D. Maurel, nous avons traité le cas des lexiques tabulaires relationnels qui permettent de coder des phénomènes linguistiques de manière factorisée, ce qui évite de dupliquer certaines contraintes, comme l'ont montré [Maurel, 1989, Constant, 2002b]. Par exemple, il est parfois possible de ramener un jeu de contraintes sur un triplet (a,b,c) d'éléments à deux jeux de contraintes indépendants sur (a,b) et (b,c). Nous supposons que les contraintes peuvent être soient positives soit négatives (matrices binaires). Dans [Constant et Maurel, 2006], nous présentons un algorithme simple et efficace combinant des opérations classiques sur les automates. A l'aide d'un automate patron, nous construisons un automate A reconnaissant toutes les séquences possibles (indépendamment des contraintes). Puis, nous construisons l'automate A^- reconnaissant les séquences interdites codées dans les tables à partir de l'automate patron. L'automate final A^+ est défini par $L(A^+) = L(A) - L(A^-)$, où L(X) correspond au langage reconnu par l'automate X. L'idée de construire l'automate des séquences interdites est intéressante car un chemin est invalide s'il subit au moins une contrainte négative. La construction de A^- est donc linéaire par rapport au nombre de contraintes. La construction de A^+ est simplement implantée au moyen d'un algorithme de type intersection. La construction directe de A^+ est très couteuse pour un système de tables complexe comme dans [Maurel, 1996], car pour qu'un chemin soit valide il faut vérifier que toutes les contraintes sont positives.

5.2.3 Transformer des lexiques tabulaires en structures de traits

La thèse d'E. Tolone [Tolone, 2011] que j'ai co-encadrée avait pour but d'exploiter les tables du lexique-grammaire dans un analyseur syntaxique symbolique (FRMG). Néanmoins, ces tables n'ont pas été conçues à la base pour être utilisées pour le TAL. Par exemple, certaines propriétés ne sont pas explicitées sauf dans la littérature (ex. propriétés définitoires des classes). Les tables ayant été conçues par différents auteurs, elles utilisent parfois des conventions de nommage de propriétés différentes, ce qui rend le lexique incohérent dans son

2. Dans lequel nous ne sommes pas impliqué.

ensemble. E. Tolone a donc réalisé un travail "archéologique" gigantesque pour remettre en ordre les tables afin qu'elles deviennent exploitables pour le TAL. Sa thèse a notamment débouché sur la distribution libre de toutes les tables sous licence LGPL-LR. Durant la période de sa thèse, nous avons particulièrement collaboré sur un point : la transformation des tables du lexique-grammaire en un lexique syntaxique dans un format XML explicite pour le TAL. Nous avons développé l'outil LGExtract qui est décrit dans [Constant et Tolone, 2010b]. Il est basé sur l'ensemble des tables du lexique-grammaire et une table des classes qui permet d'expliciter les propriétés constantes de chacune des classes qui n'étaient, à la base, pas codées. Nous avons mis au point un petit langage permettant d'associer des propriétés codées dans les tables à des règles de transformation. Ces règles permettent de rassembler toutes les informations concernant une entrée lexicale dans une même struture de traits. Nous avons généré le lexique LGlex qui est une traduction neutre des tables du lexique-grammaire. Ce lexique a servi de point de départ pour la création d'un lexique au format Lefff compatible avec l'analyseur FRMG. Un travail est actuellement en cours pour transformer LGLex au format standard ISO LMF [Francopoulo et al., 2006]. Un chapître de livre est d'ailleurs en cours de rédaction sur le sujet.

Chapitre 6

Conclusion

6.1 Bilan

Dans ce mémoire d'Habilitation à Diriger des Recherches, nous avons synthétisé et mis en perspective les différentes recherches que nous avons menées depuis nos débuts en thèse jusqu'à aujourd'hui. Nous avons essentiellement insisté sur nos travaux post-thèse depuis 2003.

Tout au long de cette période, nous avons été guidé par deux idées directrices, relativement originales en TAL : (a) la prise en compte des expressions multi-mots dans des analyseurs linguistiques et des applications ; (b) l'exploitation intensive de ressources lexicales riches dans des modèles probabilistes et des procédures hybrides. Pour (a), nous avons adopté deux grandes stratégies. La première consiste à adapter des modèles et formalismes existants à nos besoins en terme d'identification d'expressions multi-mots (ex. utilisation du schéma d'annotation BIO pour l'étiquetage morphosyntaxique). La deuxième consiste à utiliser des prétraitements pour pré-identifier des expressions multi-mots et ainsi les considérer comme des mots simples dans nos modèles et procédures. Pour (b), nous avons intégré les ressources lexicales de différentes manières : (1) limitation de l'espace de recherche en ne gardant que les analyses compatibles, (2) ajout de traits exogènes dans des modèles probabilistes, et (3) modification des symboles des grammaires. Ces deux axes sont intimement liés car les expressions multi-mots sont difficilement prédictibles et donc difficilement identifiables sans l'aide de ressources lexicales. En particulier, nous avons montré le grand intérêt des lexiques morphologiques et des ressources d'expressions multi-mots pour la reconnaissance des mots composés couplée à différents analyseurs linguistiques. Nous avons aussi montré que l'exploitation des lexiques syntaxiques dans des analyseurs probabilistes permettait d'améliorer les performances, mais de manière moindre que des méthodes non supervisées qui sont beaucoup moins couteuses. Il nous semble donc que l'exploitation de lexiques syntaxiques tels que les tables du lexique-grammaire se révélera plus pertinente pour faire le lien avec l'analyse sémantique.

Nos travaux de recherche ont été particulièrement variés, allant de la construction de ressources linguistiques à leur exploitation. Les processus hybrides que nous avons conçus intègrent des ressources lexicales riches (dictionnaires morphologiques et grammaires locales fortement lexicalisées) avec différentes technologies informatiques (ex. machines à états finis, analyseurs PCFG, etc.) et modèles (modèles probabilistes discriminants, bases de données relationnelles, etc.). Pour réaliser ces travaux, nous avons mis en place des collaborations fructueuses avec divers partenaires provenant de différents horizons (analyse syntaxique probabiliste, apprentissage, linguistique, traitement de l'oral, etc.). L'avancement de nos travaux a été relativement important ces dernières années grâce au co-encadrement de doctorants avec E. Laporte. Deux de ces thèses (celles d'Elsa Tolone soutenue en 2011 et celle de Myriam Rakho à soutenir en 2013) ont permis de faire progresser le projet collectif de notre équipe de recherche en terme de ressources lexicales. La thèse d'Anthony Sigogne (à soutenir fin 2012) est au coeur de nos problématiques scientifiques et a été un facteur très important de l'évolution des travaux décrits dans ce mémoire.

6.2 Un projet d'avenir

Dans cette section, nous présentons succintement les directions de recherche que nous souhaitons développer dans le futur.

A court terme, nous voudrions continuer certains chantiers en cours. Au niveau de l'analyse de surface, le cadre général mis en place avec des bases de données relationnelles mériterait d'être approfondi dans ses applications. Nous souhaiterions en particulier incorporer dans nos modèles des traits tenant compte des contraintes de l'automate du texte TFST généré par l'analyse lexicale et potentiellement élagué en partie par des modules de levée d'ambiguité. Un autre point à étudier est l'adaptation à différents domaines. L'utilisation de ressources lexicales devraient aider à résoudre partiellement le problème des mots simples ou composés inconnus dans le corpus d'apprentissage. Au niveau de l'analyse en profondeur, il nous semble que la méthode de réordonnancement mériterait d'être approfondie au vu des résultats Oracle montrés dans ce document. En particulier, la combinaison entre traits non-locaux généraux et traits spécifiques aux mots composés pourrait être testée au moyen d'une séquence de réordonnanceurs.

Jusqu'à présent, nous nous sommes essentiellement concentré sur les expressions multi-mots contigues de niveau lexical comme les mots composés. A plus long terme, nous souhaiterions étendre nos études aux constructions de niveau syntaxique comme les phrases figées ou les constructions à verbe support. En premier lieu, il convient de développer des ressources linguistiques pour le français ou l'anglais qui serviront de fondement à l'intégration de la reconnaissance de telles expressions dans des analyseurs syntaxiques. A notre connaissance, il n'existe pas de corpus annotés pour ces deux langues couvrant exhaustivement ce type d'expressions. Il conviendrait donc d'en créer un ou d'en compléter un, et de le mettre librement à la disposition de la communauté. Par

ailleurs, nous souhaiterions mettre en place des ressources lexicales pour ce type d'expressions exploitables pour le TAL. L'idée est de compiler diverses sources existantes, les rendre cohérentes en formalisant et explicitant entièrement leurs propriétés morphologiques et syntaxiques. Nos sources pourraient être les tables du lexique-grammaire des phrases figées (plus de 30 000 entrées) ou les dictionnaires en-ligne comme wiktionary. Nous voulons aussi développer des techniques d'acquisition automatique des expressions figées à partir du web. Pour cela, nous utiliserions certains critères linguistiques de définition (implémentables) en plus des critères statistiques classiques. Cet axe de développement de ressources linguistiques serait alors complété par un axe correspondant à l'exploitation de ces ressources pour l'intégration de la reconnaissance globale des expressions multi-mots dans des analyseurs syntaxiques. Nous souhaiterions évaluer les deux grandes approches proposées dans le mémoire : (1) pré-identification étendue des expressions multi-mots à la [Gross et Senellart, 1998] et simplification de ces expressions pour réduire la complexité lexicale dans les analyseurs ; (2) Utilisation d'un réordonnanceur d'analyses syntaxiques intégrant entre autres des traits liés aux expressions multi-mots. Par ailleurs, il nous paraît essentiel d'étendre nos travaux aux analyseurs en dépendances.

Ce travail de grande ampleur ne peut pas être le fruit d'un travail solitaire et isolé car il est très coûteux (surtout pour le développement de ressources linguistiques) et il peut avoir un impact sur toute la communauté. C'est pour cela que nous souhaiterions nous appuyer sur une (potentielle) action européenne COST que sont en train de soumettre Adam Przepiórkowski (Univ. de Varsovie), Yannick Parmentier (Univ. d'Orléans), Agata Savary (Univ. de Tours) et qui implique la plupart des chercheurs européens en TAL s'intéressant aux expressions multi-mots. Par ailleurs, le montage d'un ou plusieurs projets (ex. ANR) nous semble indispensable pour financer ce projet. Ces projets seraient également l'occasion d'appliquer nos travaux sur les expressions multi-mots à la traduction automatique en se rapprochant de partenaires spécialistes de cette tâche. L'obtention d'une HDR nous permettrait d'encadrer directement des doctorants sur des thématiques directement liées à ce projet ambitieux.

Bibliographie

[Abeillé et al., 2003] Abeillé, A., Clément, L. et Toussenel, F. (2003). Building a treebank for french. In Abeillé, A., éditeur : Treebanks. Kluwer, Dordrecht.

[Abeillé, 1993] Abeillé, A. (1993). The flexibility of french idioms : a representation with lexicalized tree adjoining grammar. In Schenk, A. et van der Linden, E., éditeurs : Idioms. Erlbaum.

[Abney, 1991] Abney, S. P. (1991). Parsing by chunks. In Berwick, R. C., Abney, S. P. et Tenny, C., éditeurs : Principle-Based Parsing : Computation and Psycholinguistics, pages 257–278. Kluwer Academic Publishers, Dordrecht.

[Abney, 1996] Abney, S. P. (1996). Partial parsing via finite-state cascades. Natural Language Engineering, 2(4):337–344.

[Adda-Decker et al., 2003] Adda-Decker, M., Habert, B., Barras, C., Adda, G., Boula de Mareüil, P. et Paroubek, P. (2003). A disfluency study for cleaning spontaneous speech automatic transcripts and improving speech language models. In ISCA Tutorial and Research Workshop on Disfluency in Spontaneous Speech (DiSS'03), pages 67–70.

[Aggarwal et al., 2004] Aggarwal, C. C., Gates, S. C. et Yu, P. S. (2004). On using partial supervision for text categorization. IEEE Trans. Knowl. Data Eng., 16(2):245–255.

[Agirre et al., 2008] Agirre, E., Baldwin, T. et Martinez, D. (2008). Improving parsing and pp attachment performance with sense information. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08), pages 317 – 325.

[Ait-Mokhtar et Chanod, 1997] Ait-Mokhtar, S. et Chanod, J.-P. (1997). Incremental finite-state parsing. In Proceedings of the fifth Conference on Applied Natural Language Processing (ANLP'97).

[Antoine et al., 2008] Antoine, J.-Y., Mokrane, A. et Friburger, N. (2008). Automatic rich annotation of large corpus of conversational transcribed speech. In Proceedings of the 8th conference on Language Resources and Evaluation (LREC'08).

[Arun et Keller, 2005] Arun, A. et Keller, F. (2005). Lexicalization in crosslinguistic probabilistic parsing : The case of french. In Proceedings of

the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05).

[Attia et al., 2010] Attia, M., Foster, J., Hogan, D., Le Roux, J., Tounsi, L. et van Genabith, J. (2010). Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL'10), Los Angeles, CA.

[Bai et al., 2009] Bai, M.-H., You, J.-M., Chen, K.-J. et Chang, J. S. (2009). Acquiring translation equivalences of multiword expressions by normalized correlation frequencies. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'09), pages 478–486.

[Baker et al., 1998] Baker, C. F., Fillmore, C. J. et Lowe, J. B. (1998). The Berkeley Framenet Project. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING'98), pages 86–90.

[Bansal et Klein, 2011] Bansal, M. et Klein, D. (2011). Web-scale features for full-scale parsing. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11), pages 693 – 702.

[Bear et al., 1993] Bear, J., Dowding, J. et Shriberg, E. (1993). A system for labeling self-repairs in speech. Rapport technique 522, Stanford Research International.

[Benzitoun, 2004] Benzitoun, C. (2004). L'annotation syntaxique de corpus oraux constitue-t-elle un problème spécifique ? In Actes de la Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'04).

[Benzitoun et al., 2004] Benzitoun, C., Campione, E., Deulofeu, J., Henry, S., Sabio, F., Teston, S., Valli, A. et Veronis, J. (2004). L'-analyse syntaxique de l'oral : problèmes et méthode. In Journée d'étude de l'ATALA sur l'annotation syntaxique de corpus.

[Blanc et Constant, 2006] Blanc, O. et Constant, M. (2006). Outilex, a linguistic platform for text processing. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pages 73–76.

[Blanc et al., 2010] Blanc, O., Constant, M., Dister, A. et Watrin, P. (2010). Partial parsing of spontaneous spoken french. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10).

[Blanc et al., 2007] Blanc, O., Constant, M. et Watrin, P. (2007). Segmentation in super-chunks with a finite-state approach. In Proceedings of the Workshop on Finite-State Methods for Natural Language Processing (FSMNLP'07).

[Blanc et al., 2006] Blanc, O., Constant, M. et Éric Laporte (2006). Outilex, plate-forme logicielle de traitement de textes écrits. In Actes du 13ème Colloque sur le traitement automatique des langues naturelles (TALN'06), pages 83 – 92.

[Blanche-Benveniste, 1985] Blanche-Benveniste, C. (1985). La dénomination dans le français parlé : une interprétation pour les 'répétitions' et les 'hésitations'. Recherches sur le français parlé, 6:99–130.

[Blanche-Benveniste et al., 1990] Blanche-Benveniste, C., Bilger, M., Rouget, C. et van den Eynde, K. (1990). Le Français parlé. Études grammaticales. CNRS Éditions.

[Blanche-Benveniste et Jeanjean, 1987] Blanche-Benveniste, C. et Jeanjean, C. (1987). Le Français parlé. Transcription et édition. Didier Érudition.

[Boullier et Sagot, 2005] Boullier, P. et Sagot, B. (2005). Analyse syntaxique profonde à grande échelle : Sxlfg. Traitement Automatique des Langues (T.A.L.), 46(2):65 – 89.

[Bouraoui et Vigouroux, 2009] Bouraoui, J.-L. et Vigouroux, N. (2009). Traitement automatique de disfluences dans un corpus linguistiquement contraint. In Actes de la conférence sur le Traitement Automatique des Langues Naturelles (TALN'09).

[Bourdaillet et al., 2010] Bourdaillet, J., Huet, S., Langlais, P. et Lapalme, G. (2010). Transsearch : from a bilingual concordancer to a translation finder. Machine Translation, 24(3-4):241–271.

[Brants, 2000] Brants, T. (2000). TNT - a statistical part-of-speech tagger. In Proceedings of the conference on Advances on Natural Language Processing (ANLP'00), pages 224 – 231.

[Brill, 1995] Brill, E. (1995). Transformation-based error-driven learning and natural language processing : A case study in part-of-speech tagging. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95), pages 543–565.

[Brin et Page, 1998] Brin, S. et Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In Proceedings of the Seventh International World-Wide Web Conference (WWW'98).

[Brown et al., 1993] Brown, P. F., Stephen, A., Pietra, A. D., Pietra, V. J. D. et Mercer, R. L. (1993). The mathematics of statistical machine translation : Parameter estimation. Computational Linguistics, 19(2):263–311.

[Brun, 1998] Brun, C. (1998). Terminology finite-state preprocessing for computational LFG. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING'98), pages 196–200.

[Burger et al., 2007] Burger, H., Dobrovol'skij, D., Kühn, P. et Norrick, N., éditeurs (2007). Phraseology : An International Handbook of Contemporary Research. Mouton de Gruyter.

[Bénard, 2005] Bénard, F. (2005). Normalisation de corpus oraux des métadonnées à l'annotation des transcriptions. Mémoire de maitrise, Université Paris 3, Sorbonne Nouvelle.

[Cafferkey et al., 2007] Cafferkey, C., Hogan, D. et van Genabith, J. (2007). Multi-word units in treebank-based probabilistic parsing and generation. In Proceedings of the 10th International Conference on Recent Advances in Natural Language Processing (RANLP'07).

[Candito et al., 2010] Candito, M., Crabbé, B. et Denis, P. (2010). Statistical french dependency parsing : treebank conversion and first results. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10).

[Candito et Crabbé, 2009] Candito, M. H. et Crabbé, B. (2009). Improving generative statistical parsing with semi-supervised word clustering. In Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09).

[Carroll et Fang, 2004] Carroll, J. et Fang, A. C. (2004). The automatic acquisition of verb subcategorisations and their impact on the performance of an hpsg parser. In Proceedings of the 1st International Conference on Natural Language Processing.

[Caseli et al., 2009] Caseli, H., Ramisch, C., Nunes, M. et Villavicencio, A. (2009). Alignment-based extraction of multiword expressions. Language resources and evaluation.

[Caseli et al., 2010] Caseli, H. d. M., Ramisch, C., das Gracas Volpe Nunes, M. et Villavicencio, A. (2010). Alignment-based extraction of multiword expressions. Language Resources and Evaluation, 44(1-2):59 – 77.

[Charniak, 1997] Charniak, E. (1997). Statistical parsing with a context-free grammar and word statistics. In Proceedings of 40th National Conference on Artificial Intelligence (AAAI'97), pages 598–603.

[Charniak, 2000] Charniak, E. (2000). A maximum-entropy-inspired parser. In Proceedings of North American Chapter of the Association for Computational Linguistics Conference (NAACL'00).

[Charniak et Johnson, 2005] Charniak, E. et Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05).

[Collins, 2000] Collins, M. (2000). Discriminative reranking for natural language parsing. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00).

[Collins, 2003] Collins, M. (2003). Head-driven statistical models for natural language parsing. Computational Linguistics, 29(4).

[Constant, 2000] Constant, M. (2000). Description d'expressions numériques en français. Revue Informatique et Statistique dans les Sciences humaines, 36:119 – 136.

[Constant, 2002a] Constant, M. (2002a). Methods for constructing lexicon-grammar resources : the example of measure expressions. In Proceedings

of the 3rd International Conference on Language Resources and Evaluation (LREC'02), pages 1341 – 1345.

[Constant, 2002b] Constant, M. (2002b). On the analysis of locative prepositional phrases : the classifier/proper noun pairing. In Proceedings of the third International Conference on Advances in Natural Language Processing (PorTAL'02), volume 2389 de Lecture Notes in Artificial Intelligence, pages 33 – 42. Springer-Verlag.

[Constant, 2003] Constant, M. (2003). Grammaires locales pour l'analyse automatique de textes : méthodes de construction et outils de gestion. Thèse de doctorat, Université Paris-Est Marne-la-Vallée.

[Constant, 2009] Constant, M. (2009). Microsyntax of measurement phrases in french : Construction and evaluation of a local grammar. In Proceedings of the 8th International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP'09).

[Constant, 2010] Constant, M. (2010). Prépositions locatives et noms propres géographiques. In Nakamura, T., Éric Laporte, Dister, A. et Fairon, C., éditeurs : Mélanges en hommage à Christian Leclère. Les Tables. La grammaire du français par le menu, Cahiers du CENTAL, pages 73 – 80. Presses Universitaires de Louvain.

[Constant, subm] Constant, M. (subm). Accounting for contiguous multiword expressions in shallow parsing. Prague Bulletin of Mathematical Linguistics, Soumis.

[Constant et Dister, 2010] Constant, M. et Dister, A. (2010). Automatic detection of disfluencies in speech transcriptions. In Pettorino, M., Giannini, A., Chiari, I. et Dovetto, F. M., éditeurs : Spoken Communication, pages 259 – 272. Cambridge Scholars Publishing.

[Constant et Dister, 2012] Constant, M. et Dister, A. (2012). Les disfluences dans les mots composés. In Actes des Journées d'Analyse de Données Textuelles (JADT'12).

[Constant et al., 2013] Constant, M., Dister, A. et Nakamura, T. (2013). Le verbe *faire* dans un corpus de français parlé. In Doa, F. K., éditeur : Penser le Lexique-Grammaire, perspectives actuelles, pages 73 – 80. Éditions Honoré Champion.

[Constant et Guglielmo, 2010] Constant, M. et Guglielmo, D. (2010). Automatic extraction of support verb construction translations from an italian-english parallel corpus. In Proceedings of the 29th International Conference on Lexis and Grammar (LGC'10), pages 63–72.

[Constant et al., subm] Constant, M., Le Roux, J. et Sigogne, A. (subm). Combining compound recognition and PCFG-LA parsing with word lattices and conditional random fields. ACM Journal of Speech and Language Processing, Soumis.

[Constant et Maurel, 2006] Constant, M. et Maurel, D. (2006). Compiling linguistic constraints into finite state automata. In Proceedings of the 11th

International Conference on Implementation and Application of Automata (CIAA'06), volume 4094 de Lecture Notes in Computer Science. Springer.

[Constant et al., 2010] Constant, M., Nakamura, T., Voyatzi, S. et Bittar, A. (2010). Extraction automatique de traductions anglaises de mots composés français. In Actes du Congrés Mondial de la Linguistique Française (CMLF'10).

[Constant et Sigogne, 2011] Constant, M. et Sigogne, A. (2011). MWU-aware part-of-speech tagging with a CRF model and lexical resources. In Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World (MWE'11).

[Constant et al., 2012a] Constant, M., Sigogne, A. et Watrin, P. (2012a). Discriminative strategies to integrate multiword expression recognition and parsing. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12), pages 204–212.

[Constant et al., 2012b] Constant, M., Sigogne, A. et Watrin, P. (2012b). La reconnaissance des mots composés à l'épreuve de l'analyse syntaxique et vice-versa : évaluation de deux stratégies discriminantes. In Actes de la conférence sur le Traitement Automatique des Langues Naturelles (TALN'12), pages 57–70.

[Constant et Tellier, 2012] Constant, M. et Tellier, I. (2012). Evaluating the impact of external lexical resources into a crf-based multiword segmenter and part-of-speech tagger. In Proceedings of the 8th conference on Language Resources and Evaluation (LREC'12).

[Constant et Tellier, subm] Constant, M. et Tellier, I. (subm). How to integrate external knowledge into a learning process application to joint multiword unit recognition and part-of-speech tagging. Linguistic Issues in Language Technology, Soumis.

[Constant et al., 2011] Constant, M., Tellier, I., Duchier, D., Dupont, Y., Sigogne, A. et Billot, S. (2011). Intégrer des connaissances linguistiques dans un crf : application à l'apprentissage d'un segmenteur-étiqueteur du français. In Actes de Conférence sur le traitement automatique des langues naturelles (TALN'11).

[Constant et Tolone, 2010a] Constant, M. et Tolone, E. (2010a). A generic tool to generate a lexicon for NLP from lexicon-grammar tables. In Gioia, M. D., éditeur : Actes du 27e Colloque international sur le lexique et la grammaire. Seconde partie, pages 79–93. Aracne.

[Constant et Tolone, 2010b] Constant, M. et Tolone, E. (2010b). A generic tool to generate a lexicon for nlp from lexicon-grammar tables. In Actes du 27e Colloque international sur le lexique et la grammaire. Seconde partie, pages 79 – 93. Aracne.

[Constant et Watrin, 2008] Constant, M. et Watrin, P. (2008). Networking multiword units. In Nordstrom, B. et Ranta, A., éditeurs : Proceedings of the 6th International Conference on Natural Language Processing

(GoTAL'08), volume 5221 de Lecture Notes in Artificial Intelligence, pages 120 – 125. Springer-Verlag.

[Copestake et al., 2002] Copestake, A., Lambeau, F., Villavicencio, A., Bond, F., Baldwin, T., Sag, I. et Flickinger, D. (2002). Multiword expressions : Linguistic precision and reusability. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02).

[Courtois, 2009] Courtois, B. (2009). Un système de dictionnaires électroniques pour les mots simples du français. Langue Française, 87:1941 – 1947.

[Courtois et al., 1997] Courtois, B., Garrigues, M., Gross, G., Gross, M., Jung, R., Mathieu-Colas, M., Monceaux, A., Poncet-Montange, A., Silberztein, M. et Vivés, R. (1997). Dictionnaire électronique DELAC : les mots composés binaires. Rapport technique 56, LADL, University Paris 7.

[Cowie, 1998] Cowie, A. P., éditeur (1998). Phraseology : Theory, Analysis, and Application. Clarendon Press.

[Crabbé, 2005] Crabbé, B. (2005). Représentation informatique de grammaires d'arbres fortement lexicalisées : le cas de la grammaire d'arbres adjoints. Thèse de doctorat, Université Nancy 2.

[Crabbé et Candito, 2008] Crabbé, B. et Candito, M. H. (2008). Expériences d'analyse syntaxique statistique du français. In Actes de TALN 2008 (Traitement automatique des langues naturelles), Avignon. ATALA.

[D'Agostino et Vietri, 2004] D'Agostino, Emilio, A. E. et Vietri, S. (2004). Lexicon-grammar, electronic dictionaries and local grammars of italian. In Lexique, syntaxe et lexique-grammaire. Papers in honour of Maurice Gross, volume 4 de Lingvisticae Investigationes Supplementa, pages 125 – 136. Benjamins.

[Daille, 1995] Daille, B. (1995). Repérage et extraction de terminologie par une approche mixte statistique et linguistique. traitement Automatique des Langues (TAL), 36(1-2):101–118.

[Danlos, 1980] Danlos, L. (1980). Représentation d'informations linguistiques : les constructions N être Prep X. Thèse de doctorat, Université Paris 7.

[Danlos, 1994] Danlos, L. (1994). Coder des informations monolingues sur les noms pour éviter des règles bilingues sensibles au contexte. Langages, 28(116).

[Danlos, 2009] Danlos, L. (2009). Extension de la notion de verbe support. Support et prédicats non verbaux dans les langues du monde.

[Danlos et al., 2006] Danlos, L., Sagot, B. et Salmon-Alt, S. (2006). French frozen verbal expressions : from lexicon-grammar tables to nlp applications. In Proceedings of the Lexicon Grammar Conference (LGC'06).

[Debusmann, 2004] Debusmann, R. (2004). Multiword expressions as dependency subgraphs. In Proceedings of the ACL Workshop on Multiword Expressions : Integrating Processing.

[Deléger et al., 2009] Deléger, L., Merkel, M. et Zweigenbaum, P. (2009). Translating medical terminologies through word alignment in parallel text corpora. Journal of Biomedical Informatics, 42:692–701.

[Denis et Sagot, 2009] Denis, P. et Sagot, B. (2009). Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC'09).

[Deoskar, 2008] Deoskar, T. (2008). Re-estimation of lexical parameters for treebank PCFGs. In Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08), pages 193 – 200.

[Dias, 2003] Dias, G. (2003). Multiword unit hybrid extraction. In Proceedings of the Workshop on Multiword Expressions of the 41st Annual Meeting of the Association of Computational Linguistics (MWE'03), pages 41–49.

[Dister, 2007] Dister, A. (2007). De la transcription à l'étiquetage morphosyntaxique. Le cas de la banque de données textuelles orales Valibel. Thèse de doctorat, Université de Louvain.

[Dister et al., 2009a] Dister, A., Constant, M. et Prunelle, G. (2009a). Normalizing speech transcriptions for natural language processing. In Proceedings of the international conference on Spoken Communication.

[Dister et al., 2009b] Dister, A., Francard, M., Hambye, P. et Simon, A. (2009b). Du corpus à la banque de données. du son, des textes et des métadonnées. l'évolution de banque de données textuelles orales VALIBEL (1989-2009). Cahiers de Linguistique, 33(2):113–129.

[Dubois et Dubois-Charlier, 1997] Dubois, J. et Dubois-Charlier, F. (1997). Dictionnaire des verbes français. Larousse.

[Duchier et Debusmann, 2001] Duchier, D. et Debusmann, R. (2001). Topological dependency trees : A constraint-based account of linear precedence. In Proceedings of the 39th Meeting of the Association for Computational Linguistics (ACL'01).

[Dunning, 1993] Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. Computational Linguistics, 19(1):61–74.

[Ehrmann, 2008] Ehrmann, M. (2008). Les Entités Nommées, de la linguistique au TAL, statut théorique et méthodes de désambiguïsation. Thèse de doctorat, Université Paris 7.

[Erkan et Radev, 2004] Erkan, G. et Radev, D. R. (2004). Lexrank : graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, 22(1).

[Eryigit et al., 2011] Eryigit, G., Ilbay, T. et Arkan Can, O. (2011). Multiword expressions in statistical dependency parsing. In Proceedings of the IWPT Workshop on Statistical Parsing of Morphologically-Rich Languages (SPRML'11).

[Eshkol et al., 2010] Eshkol, I., I., T., Taalab, S. et Billot, S. (2010). Étiqueter un corpus oral par apprentissage automatique à l'aide de connaissances linguistiques. In Actes des Journées d'Analyse de Données Textuelles (JADT'10).

[Federici et al., 1996] Federici, S., Montemagni, S. et Pirelli, V. (1996). Shallow parsing and text chunking : A view on underspecification in syntax. In Proceedings of the ESSLLI'96 Workshop on Robust Parsing.

[Fiala et al., 1997] Fiala, P., Lafin, P. et Piguet, M.-F., éditeurs (1997). Locution : entre lexique, syntaxe et pragmatique. Klincksieck.

[Fillmore et al., 2003] Fillmore, C. J., Johnson, C. R. et Petruck, M. R. (2003). Background to framenet. International Journal of Lexicography, 16(3).

[Finkel et Manning, 2009] Finkel, J. R. et Manning, C. D. (2009). Joint parsing and named entity recognition. In Proceedings of the conference of the North American Chapter of the Association for Computational Linguistics (NAACL'09).

[Francopoulo et al., 2006] Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M. et Soria, C. (2006). Lexical markup framework (lmf). In Proceedings of the Linguistic Resources and Evaluation Conference (LREC'06).

[Garside, 1995] Garside, R. (1995). Grammatical tagging of the spoken part of the british national corpus : a progress report. In Leech, G., Myers, G. et Thomas, J., éditeurs : Spoken English on Computer. Transcription, Mark-up and Application, pages 161–167. Longman.

[Gerdes et Kahane, 2001] Gerdes, K. et Kahane, S. (2001). Word order in german : A formal dependency grammar using a topological hierarchy. In Proceedings of the 39th Meeting of the Association for Computational Linguistics (ACL'01).

[Gimenez et Marquez, 2004] Gimenez, J. et Marquez, L. (2004). Svmtool : A general pos tagger generator based on support vector machines. In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04).

[Giry-Schneider, 1978] Giry-Schneider, J. (1978). Les nominalisations en français : l'opérateur 'faire' dans le lexique. Droz.

[Giry-Schneider, 1987] Giry-Schneider, J. (1987). Les prédicats nominaux en français : les phrases simples à verbe support. Droz.

[Giry-Schneider, 1991] Giry-Schneider, J. (1991). Noms de grandeur en avoir et noms d'unité. Cahiers de grammaire, 16.

[Green et al., 2011] Green, S., de Marneffe, M.-C., Bauer, J. et Manning, C. D. (2011). Multiword expression identification with tree substitution grammars : A parsing tour de force with french. In Proceedings of the conference on Empirical Method for Natural Language Processing (EMNLP'11).

[Grishman et al., 1994] Grishman, R., Macleod, C. et Meyers, A. (1994). COMLEX syntax : building a computational lexicon. In Proceedings of the conference on Computational Linguistics (COLING'94).

[Grishman et Sundheim, 1996] Grishman, R. et Sundheim, B. (1996). Message understanding conference - 6 : A brief history. In Proceedings of the International Conference on Computational Linguistics (COLING'96).

[Gross, 1989] Gross, G. (1989). Les constructions converses du français. Droz.

[Gross, 1996a] Gross, G. (1996a). Les expressions figées en français. Noms composés et autres locutions. Orphrys.

[Gross, 1975] Gross, M. (1975). Méthodes en syntaxe : régimes des constructions complétives. Hermann, Paris, France.

[Gross, 1981] Gross, M. (1981). Les bases empiriques de la notion de prédicat sémantique. Langages, 63:7–52.

[Gross, 1982] Gross, M. (1982). Une classification des phrases "figées" du français. Revue québécoise de linguistique, 11(2):151 – 185.

[Gross, 1986] Gross, M. (1986). Lexicon grammar. the representation of compound words. In Proceedings of Computational Linguistics (COLING'86).

[Gross, 1988] Gross, M. (1988). Les limites de la phrase figée. Langages, 90:7 – 22.

[Gross, 1994] Gross, M. (1994). Constructing Lexicon-Grammars, pages 213–263. Oxford University Press, Oxford.

[Gross, 1996b] Gross, M. (1996b). Les formes être prép x du français. Lingvisticae Investigationes, 20(2).

[Gross, 1997] Gross, M. (1997). The construction of local grammars. In Roche, E. et Schabes, Y., éditeurs : Finite-State Language Processing, pages 329–352. The MIT Press, Cambridge, Mass.

[Gross, 1999a] Gross, M. (1999a). Lemmatization of compound tenses in english. Lingvisticae Investigationes, 22.

[Gross, 1999b] Gross, M. (1999b). Sur la définition d'auxiliaire du verbe. Langages, 135:8–21.

[Gross et Senellart, 1998] Gross, M. et Senellart, J. (1998). Nouvelles bases statistiques pour les mots du français. In Actes des Journées d'Analyse statistique des Données Textuelles (JADT'98), pages 335–150.

[Grossmann et Tutin, 2003] Grossmann, F. et Tutin, A., éditeurs (2003). Les collocations : analyse et traitement. Travaux et recherches en linguistique appliquée. De Werelt.

[Grégoire, 2008] Grégoire, N. (2008). DuELME : a dutch electronic lexicon of multiword expressions. Language Resources and Evaluation, 44(1-2).

[Guillet et Leclère, 1992] Guillet, A. et Leclère, C. (1992). La structure des phrases simples en français 2 : les constructions transitives locatives. Droz.

[Guénot, 2005] Guénot, M.-L. (2005). Parsing de l'oral : traiter les disfluences. In Actes de la conférence sur le Traitement Automatique des Langues Naturelles (TALN'05).

[Haussman, 1979] Haussman, F. (1979). Un dictionnaire des collocations est-il possible ? Travaux de Linguistique et de Littérature, 17(1):187 – 195.

[Haveliwala, 2003] Haveliwala, T. H. (2003). Topic-sensitive pagerank : A context-sensitive ranking algorithm for web search. IEEE Transactions on Knowledge and Data Engineering, 15.

[Heid, 1994] Heid, U. (1994). On ways words work together. In Research topics in lexical combinatorics. Proceedings of the 6th Euralex International Congress on Lexicography (EURALEX'94), pages 226–257.

[Hogan et al., 2011] Hogan, D., Foster, J. et van Genabith, J. (2011). Decreasing lexical data sparsity in statistical syntactic parsing - experiments with named entities. In Proceedings of ACL Workshop Multiword Expressions : from Parsing and Generation to the Real World (MWE'2011).

[Huang, 2008] Huang, L. (2008). Forest reranking : Discriminative parsing with non-local features. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08).

[Ide et Wilks, 2006] Ide, N. et Wilks, A. (2006). Making sense about sense. In Agirre, E. et Edmonds, P., éditeurs : Word Sense Disambiguation : Algorithms and Applications, pages 47–74. Springer, Cambridge, Mass.

[Johnson, 1998] Johnson, M. (1998). PCFG models of linguistic tree representations. Computational Linguistics, 24:613–632.

[Joshi et al., 1975] Joshi, A. K., Levy, L. et Takahashi, M. (1975). Tree adjunct grammars. Journal of Computer and System Science, 10(1):136 – 163.

[Kaalep et Muischnek, 2008] Kaalep, H.-J. et Muischnek, K. (2008). Multiword verbs of estonian : a database and a corpus. In Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE'08), pages 23 – 26.

[Kaplan et Bresnan, 1982] Kaplan, R. et Bresnan, J. (1982). Lexical-functional grammar : a formal system for grammatical representation. In The Mental Representation of Grammatical Relations, pages 173 – 281. MIT Press.

[Karlsson et al., 1995] Karlsson, F., Voutilainen, A., Heikkila, J. et Anttila, A. (1995). Constraint Grammar : A language-independent system for parsing unrestricted text, volume 4 de Natural Language Processing. Mouton de Gruyter.

[Karttunen, 2001] Karttunen, L. (2001). Applications of finite-state transducers in natural language processing. In Proceedings of the 5th International Conference on Implementation and Application of Automata (CIAA'00), volume 2088 de Lecture Notes in Computer Science, pages 34–46.

[Kim et Baldwin, 2007] Kim, S. N. et Baldwin, T. (2007). Interpreting noun compounds using bootstrapping and sense collocation. In Proceedings of Conference of the Pacific Association for Computational Linguistics (PACLING'07), pages 129 – 136.

[Kipper et al., 2000] Kipper, K., Dang, H. T. et Palmer, M. (2000). Class-based construction of a verb lexicon. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pages 691 – 696.

[Klein et Manning, 2003] Klein, D. et Manning, C. D. (2003). Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03).

[Koehn, 2005] Koehn, P. (2005). Europarl : A parallel corpus for statistical machine translation. In Proceedings of the Tenth machine Translation Summit (MT Summit X), pages 79 – 86.

[Koo et al., 2008] Koo, T., Carreras, X. et Collins, M. (2008). Simple semi-supervised dependency parsing. In Proceedings of the 46th Annual Meeting on Association for Computational Linguistics (ACL'08).

[Korkontzelos et Manandhar, 2010] Korkontzelos, I. et Manandhar, S. (2010). Can recognising multiword expressions improve shallow parsing ? In Proceedings of Human Language Technologies : The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'10), pages 636–644.

[Kornai, 1999] Kornai, A., éditeur (1999). Extended Finite State Models of Language. Cambridge University Press.

[Kudo et Matsumoto, 2001] Kudo, T. et Matsumoto, Y. (2001). Chunking with support vector machines. In Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'01).

[Lafferty et al., 2001] Lafferty, J., McCallum, A. et Pereira, F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01), pages 282–289.

[Laporte, 2007] Laporte, E. (2007). Extension of a grammar of French determiners. In Proceedings of the international conference on Lexicon and Grammar (LGC'07), pages 65 – 72.

[Laporte et Monceaux, 1999] Laporte, E. et Monceaux, A. (1999). Elimination of lexical ambiguities by grammars. the ELAG system. Lingvisticae Investigationes, XXII:341 – 367.

[Laporte et al., 2008] Laporte, E., Nakamura, T. et Voyatzi, S. (2008). A french corpus annotated for multiword nouns. In Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE'08), pages 23 – 26.

[Lavergne et al., 2010] Lavergne, T., Cappé, O. et Yvon, F. (2010). Practical very large scale CRFs. In Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10), pages 504–513.

[Le Roux et al., 2011] Le Roux, J., Favre, B., Mirroshandel, S. A. et Nasr, A. (2011). Modèles génératif et discriminant en analyse syntaxique : expériences sur le corpus arboré de paris 7. In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'11).

[Lendvai et al., 2003] Lendvai, P., van den Bosch, A. et Krahmer, E. (2003). Memory-based disfluency chunking. In Proceedings of Disfluency in Spontaneous Speech Workshop (DISS'03), pages 63–66.

[Levin, 1993] Levin, B. (1993). English Verb Classes and Alternations, A Preliminary Investigation. University of Chicago Press.

[Liu et al., 2006] Liu, Y., Shriberg, E., Stolcke, A., Hillard, D., Ostendorf, M. et Harper, M. (2006). Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. IEEE Transactions on Audio, Speech, and Language Processing, 14(5):1526 – 1540.

[Lü et Zhou, 2004] Lü, Y. et Zhou, M. (2004). Collocation translation acquisition using monolingual corpora. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), pages 167 – 174.

[Macleod et al., 1998] Macleod, C., Grishman, R., Meyers, A., Barrett, L. et Reeves, R. (1998). NOMLEX : A lexicon of nominalizations. In Actes du Huitième congrès international de lexicographie (EURALEX'98), pages 187–193.

[Manning et Schütze, 1999] Manning, C. D. et Schütze, H. (1999). Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge.

[Marcus et al., 1993] Marcus, M. P., Santorini, B. et Marcinkiewicz, M.-A. (1993). Building a large annotated corpus of english : The penn treebank. Computational Linguistics, 19(2).

[Martineau et al., 2009] Martineau, C., Nakamura, T., Varga, L. et Voyatzi, S. (2009). Annotation et normalisation des entités nommées. Arena Romanistica, 4:234–243.

[Matsuzaki et al., 2005] Matsuzaki, T., Miyao, Y. et Tsujii, J. (2005). Probabilistic CFG with latent annotations. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 75 – 82.

[Maurel, 1989] Maurel, D. (1989). Reconnaissance de séquences de mots par automates. Adverbes de dates du français. Thèse de doctorat, Université Paris 7.

[Maurel, 1996] Maurel, D. (1996). Building automaton on schemata and acceptability tables. In First Workshop on Implementing automata (WIA'96), volume 1260 de Lecture Notes in Computer Science, pages 72–86.

[Mcdonald et al., 2005] Mcdonald, R., Crammer, K. et Pereira, F. (2005). Online large-margin training of dependency parsers. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05), pages 91 – 98.

[Mel'cuk, 1998] Mel'cuk, I. A. (1998). Collocations and lexical functions. In Cowie, A. P., éditeur : Phraseology. Theory, Analysis and Applications, pages 23 – 53. Clarendon Press.

[Mel'cuk, 2011] Mel'cuk, I. A. (2011). Phrasèmes dans le dictionnaire. In Jean-Claude, A. et Salah, M., éditeurs : Le figement linguistique : la parole entrâvée, pages 41 – 61. Honoré Champion.

[Mel'cuk et al., 1999] Mel'cuk, I. A., Arbatchewsky-Jumarie, N., Elnitsky, L. et Lessard, A. (1984-1999). Dictionnaire explicatif et combinatoire du français contemporain : Recherches lexico-sémantiques. Volume I-IV. Presses de l'Université de Montréal.

[Messiant et al., 2008] Messiant, C., Korhonen, A. et Poibeau, T. (2008). Lexschem : A large subcategorization lexicon for french verbs. In Proceedings of the Language Resources and Evaluation Conference (LREC'08).

[Mihalcea et Radev, 2011] Mihalcea, R. et Radev, D. (2011). Graph-based Natural Language Processing and Information Retrieval. Cambridge University Press.

[Mirroshandel et al., 2012] Mirroshandel, S. A., Nasr, A. et Roux, J. L. (2012). Semi-supervised dependency parsing using lexical affinities. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12).

[Mohri, 1997] Mohri, M. (1997). Finite-state transducers in language and speech processing. Computational Linguistics, 23(2):269–311.

[Moreau et Tellier, 2009] Moreau, E. et Tellier, I. (2009). The crotal srl system : a generic tool based on treestructured CRF. In proceedings of Computational Natural Language Learning (CoNLL'09), shared task, pages 91–96.

[Morin et Daille, 2010] Morin, E. et Daille, B. (2010). Compositionality and lexical alignment of multi-word terms. Language Resources and Evaluation, 44(1/2):79 – 95.

[Nakamura et Constant, 2001] Nakamura, T. et Constant, M. (2001). Les expressions de pourcentage. Flambeau, 27:27 – 46.

[Nakov, 2008] Nakov, P. (2008). Noun compound interpretation using paraphrasing verbs : Feasibility study. In Proceedings of the 13th international conference on Artificial Intelligence : Methodology, Systems, and Applications (AIMSA'08), pages 103–117.

[Nasr et al., 2010] Nasr, A., Béchet, F. et Rey, J.-F. (2010). MACAON : Une chaîne linguistique pour le traitement de graphes de mots. In Actes de la conférence sur le Traitement Automatique des Langues Naturelles (TALN'10) - session de démonstrations.

[Nasr et Volanschi, 2005] Nasr, A. et Volanschi, A. (2005). Integrating a POS tagger and a chunker implemented as weighted finite state machines. In Proceedings of the workshop on Finite-State Methods and Natural Language Processing (FSMNLP'05), pages 167 – 178.

[Nivre, 2003] Nivre, J. (2003). An efficient algorithm for projective dependency parsing. In Proceedings of the 8th International Workshop on Parsing Technologies (IWPT'03), pages 149–160.

[Nivre et Grönqvist, 2001] Nivre, J. et Grönqvist, L. (2001). Tagging a corpus of spoken swedish. International Journal of Corpus Linguistics, 6(1):47–78.

[Nivre et al., 2006] Nivre, J., Hall, J. et Nilsson, J. (2006). Maltparser : A data-driven parser-generator for dependency parsing. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06), pages 2216 – 2219.

[Nivre et Nilsson, 2004] Nivre, J. et Nilsson, J. (2004). Multiword units in syntactic parsing. In Proceedings of Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA).

[Och et Ney, 2003] Och, F. J. et Ney, H. (2003). A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19–51.

[Oostdijk, 2003] Oostdijk, N. (2003). Normalization and disfluencies in spoken language data. In Granger, S. et Petch-Tyson, S., éditeurs : Extending the scope of corpus-based research. New applications, new challenges, pages 59–70. Rodopi.

[Paroubek et al., 2007] Paroubek, P., Vilnat, A., Robba, I. et Ayache, C. (2007). Les résultats de la campagne EASY d'évaluation des analyseurs syntaxiques du français. In Actes de la conférence sur le Traitement Automatique des Langues Naturelles (TALN'07).

[Paumier, 2003] Paumier, S. (2003). De la reconnaissance de formes linguistiques à l'analyse syntaxique. Thèse de doctorat, Université Paris-Est Marne-la-Vallée.

[Pecina, 2010] Pecina, P. (2010). Lexical association measures and collocation extraction. Language Resources and Evaluation, 44:137–158.

[Petrov et al., 2006] Petrov, S., Barrett, L., Thibaux, R. et Klein, D. (2006). Learning accurate, compact and interpretable tree annotation. In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL'06).

[Petrov, 2010] Petrov, V. (2010). Products of random latent variable grammars. In Proceedings of Human Language Technologies : The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'10).

[Piton et al., 1999] Piton, O., Maurel, D. et Belleil, C. (1999). The prolex data base : Toponyms and gentiles for nlp. In Proceedings of the Third

International Workshop on Applications of Natural Language to Data Bases (NLDB'99), pages 233–237.

[Pollard et Sag, 1994] Pollard, C. et Sag, I. (1994). Head-Driven Phrase Structure Grammar. Chicago University Press.

[Radev et al., 1999] Radev, D. R., Hatzivassiloglou, V. et McKeow, K. R. (1999). A description of the CIDR system as used for tdt-2. In Proceedings of the DARPA Broadcast News Workshop.

[Rakho et al., 2012] Rakho, M., Laporte, E. et Constant, M. (2012). A new semantically annotated corpus with syntax-based and cross-lingual senses. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12).

[Ramisch et al., 2010a] Ramisch, C., Villavicencio, A. et Boitet, C. (2010a). mwe-toolkit : a framework for multiword expression identification. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10).

[Ramisch et al., 2010b] Ramisch, C., Villavicencio, A. et Boitet, C. (2010b). Web-based and combined language models : a case study on noun compound identification. In Proceedings of the Conference on Computational Linguistics (COLING'10).

[Ramshaw et Marcus, 1995] Ramshaw, L. A. et Marcus, M. P. (1995). Text chunking using transformation-based learning. In Proceedings of the 3rd Workshop on Very Large Corpora, pages 88 – 94.

[Ranchhod et al., 2004] Ranchhod, E., Carvalho, P., Mota, C. et Barreiro, A. (2004). Portuguese large-scale language resources for nlp applications. In Proceedings of the 4th conference on Language Resources and Evaluation (LREC'04), pages 1755–1758.

[Ratnaparkhi, 1996] Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'96), pages 133 – 142.

[Riezler et al., 2002] Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell, J. T., et Johnson, M. (2002). Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02).

[Roche, 1997] Roche, E. (1997). Transducer parsing of free and frozen sentences. Natural Language Engineering, 2(4):345 – 350.

[Roche et Schabes, 1995] Roche, E. et Schabes, Y. (1995). Deterministic part-of-speech tagging with finite-state transducers. Computational Linguistics, 21(2):227 – 253.

[Sag et al., 2002] Sag, I. A., Baldwin, T., Bond, F., Copestake, A. A. et Flickinger, D. (2002). Multiword expressions : A pain in the neck for nlp. In Proceedings of the Third International Conference on Computational

Linguistics and Intelligent Text Processing (CICLing'02), pages 1–15, London, UK. Springer-Verlag.

[Sagot, 2010] Sagot, B. (2010). The lefff, a freely available, accurate and large-coverage lexicon for french. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10).

[Samardzic et Merlo, 2010] Samardzic, T. et Merlo, P. (2010). Cross-lingual variation of light verb constructions : using parallel corpora and automatic alignment for linguistic research. In Proceedings of the ACL Workshop on NLP and Linguistics : Finding the Common Ground, pages 52 – 60.

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

[Schmid, 1995] Schmid, H. (1995). Probabilistic part-of-speech tagging using decision trees. In Proceedings of the International Conference on New Methods in Language Processing, (ICNMLP'95),.

[Schuler et Joshi, 2011] Schuler, W. et Joshi, A. (2011). Tree-rewriting models of multi-word expressions. In Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World (MWE'11).

[Seddah et al., 2009] Seddah, D., Candito, M.-H. et Crabbé, B. (2009). Cross-parser evaluation and tagset variation : a french treebank study. In Proceedings of International Workshop on Parsing Technologies (IWPT'09).

[Sekine et al., 2002] Sekine, S., Sudo, K. et Nobata, C. (2002). Extended named entity hierarchy. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02).

[Seretan et al., 2003] Seretan, V., Nerima, L. et 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'03), pages 424–431.

[Seretan et Wehrli, 2007] Seretan, V. et Wehrli, E. (2007). Collocation translation based on sentence alignment and parsing. In Actes de la conférence sur le Traitement Automatique des Langues Naturelles (TALN'07).

[Sha et Pereira, 2003] Sha, F. et Pereira, F. (2003). Shallow parsing with conditional random fields. In Proceedings of the Conference on Human Language Technologies and the Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03), pages 213 – 220.

[Sigogne, 2009] Sigogne, A. (2009). De l'étiquetage morpho-syntaxique au super-chunking : Levée d'ambiguïtés à l'aide de méthodes hybrides et de ressources lexicales riches. Mémoire de stage de Master 2, Université Paris-Est Marne-la-Vallée.

[Sigogne, 2010] Sigogne, A. (2010). Hybridtagger : un étiqueteur hybride pour le français. In Actes de la 8ème MAnifestation des JEunes Chercheurs

en Sciences et Technologies de l'Information et de la Communication (MajecSTIC'10).

[Sigogne, 2012] Sigogne, A. (2012). Intégration de ressources lexicales riches dans un analyseur syntaxique probabiliste. Thèse de doctorat, Université Paris-Est.

[Sigogne et Constant, 2009] Sigogne, A. et Constant, M. (2009). Real-time unsupervised classification of web documents. In Proceedings of the 4th International Multiconference on Computer Science and Information Technology (IMCSIT'09), pages 281 – 286.

[Sigogne et Constant, 2012] Sigogne, A. et Constant, M. (2012). Using sub-categorization frames to improve french probabilistic parsing. In Proceedings of the 11th Conference on Natural Language Processing (KONVENS'12).

[Sigogne et al., 2011a] Sigogne, A., Constant, M. et Éric Laporte (2011a). French parsing enhanced with a word clustering method based on a syntactic lexicon. In Second Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL'11).

[Sigogne et al., 2011b] Sigogne, A., Constant, M. et Éric Laporte (2011b). Integration of data from a syntactic lexicon into a generative and a discriminative probabilistic parsers. In Proceedings of the International conference on Recent Advances in Natural Language Processing (RANLP'11).

[Silberztein, 1994] Silberztein, M. (1994). Intex : a corpus processing system. In Proceedings of the conference on Computational Linguistics (COLING'94).

[Silberztein, 2000] Silberztein, M. (2000). INTEX : an FST toolbox. Theoretical Computer Science, 231(1):33–46.

[Silberztein, 2003] Silberztein, M. (2003). Finite-state description of the french determiner system. Journal of French Language Studies, 13(2).

[Silberztein, 2008] Silberztein, M. (2008). Complex annotations with NooJ. In Proceedings of the International NooJ Conference. Cambridge Scholars Publishing.

[Simard, 2003] Simard, M. (2003). Translation spotting for translation memories. In Proceedings of HLT-NAACL Workshop on Building and Using Parallel Texts : Data Driven Machine translation and Beyond, pages 65–72.

[Sinclair, 1991] Sinclair, J. (1991). Corpus, Concordance, Collocations. Oxford University Press.

[Smadja et al., 1996] Smadja, F., McKeown, K. et Hatzivassiloglou (1996). Translating collocations for bilingual lexicons : a statistical approach. Computational Linguistics, 22(1):1–38.

[Smadja, 1993] Smadja, F. A. (1993). Retrieving collocations from text : Xtract. Computational Linguistics, 19(1):143 – 177.

[Snover et al., 2004] Snover, M., Dorr, B. et Schwartz, R. (2004). A lexically-driven algorithm for disfluency detection. In Proceedings of the Conference of the North American Chapter of the Association

for Computational Linguistics : Human Language Technologies (HLT-NAACL'04), pages 157–160.

[Suzuki et al., 2009] Suzuki, J., Isozaki, H., Carreras, X. et Collins, M. (2009). An empirical study of semisupervised structured conditional models for dependency parsing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'09), pages 551 – 560.

[Thede et Harper, 1999] Thede, S. et Harper, M. (1999). A second-order hidden markov model for part-of-speech tagging. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'99), pages 175 – 182.

[Thomasset et de La Clergerie, 2005] Thomasset, F. et de La Clergerie, E. (2005). Comment obtenir plus des métagrammaires. In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'05).

[Tolone, 2011] Tolone, E. (2011). Analyse syntaxique à l'aide des tables du Lexique-Grammaire du français. Thèse de doctorat, Université Paris-Est.

[Toutanova et al., 2003] Toutanova, K., Klein, D., Manning, C. et Singer, Y. Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of Human Language Technologies : The 4th Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03), pages 252 – 259.

[Tsuruoka et al., 2009] Tsuruoka, Y., Tsujii, J. et Ananiadou, S. (2009). Fast full parsing by linear-chain conditional random fields. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL'09), pages 790–798.

[Tu et Roth, 2011] Tu, Y. et Roth, D. (2011). Learning english light verb constructions : Contextual or statistical. In Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World, pages 31–39.

[Tutin et Grossmann, 2002] Tutin, A. et Grossmann, F. (2002). Collocations régulières et irrégulières : esquisse de typologie du phénomène collocatif. Revue Française de Linguistique Appliquée, Lexique : recherches actuelles, 7:7 – 25.

[Valli et Véronis, 1999] Valli, A. et Véronis, J. (1999). Étiquetage grammatical des corpus de parole : problèmes et perspectives. Revue française de linguistique appliquée, 4(2):113–133.

[Van den Eynde et Mertens, 2003] Van den Eynde, K. et Mertens, P. (2003). La valence : l'approche pronominale et son application au lexique verbal. Journal of French Language Studies, 13:63–104.

[Veronis, 2003] Veronis, J. (2003). Hyperlex : cartographie lexicale pour la recherche d'information. In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'03), pages 265–275.

[Vincze et al., 2011a] Vincze, V., Nagy, I. et Berend, G. (2011a). Detecting noun compounds and light verb constructions : a contrastive study. In

Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World (MWE'11), pages 116 – 121.

[Vincze et al., 2011b] Vincze, V., Nagy, I. et Berend, G. (2011b). Multiword expressions and named entities in the wiki50 corpus. In Proceedings of the conference on Recent Advances in Natural Language Processing (RANLP'11), pages 289–295.

[Watrin, 2006] Watrin, P. (2006). Une approche hybride de l'extraction d'information : sous-langages et lexique-grammaire. Thèse de doctorat, Université catholique de Louvain.

[Watrin et François, 2011] Watrin, P. et François, T. (2011). N-gram frequency database reference to handle mwe extraction in nlp applications. In Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World (MWE'11).

[Wehrli et al., 2010] Wehrli, E., Seretan, V. et Nerima, L. (2010). Sentence analysis and collocation identification. In Proceedings of the Workshop on Multiword Expression : From Theory to Applications (MWE'10).

[Wehrli et al., 2009] Wehrli, E., Seretan, V., Nerima, L. et Russo, L. (2009). Collocations in a rule-based mt system : A case study evaluation of their translation adequacy. In Proceedings of the 13th Meeting of the European Association for Machine Translation (EAMT'09), pages 128 – 135.

[Woods, 1970] Woods, W. (1970). Transition network grammars for natural language analysis. Communications of the ACM, 13(10).

[Yamada et Matsumoto, 2003] Yamada, H. et Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In Proceedings of the 8th International Workshop on Parsing Technologies (IWPT'03), pages 195–206.

[Zarrieß et Kuhn, 2009] Zarriess, S. et Kuhn, J. (2009). Exploiting translational correspondences for pattern-independent mwe identification. In Proceedings of the Workshop on Multiword Expressions : Identification, Interpretation, Disambiguation and Applications (MWE'09), pages 23 – 30.