Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français

Matthieu Constant, Isabelle Tellier, Denys Duchier, Yoann Dupont, Anthony Sigogne, Sylvie Billot

TALN 2011, Montpellier, 27 *juin* –1*er juillet 2011*

Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français

Matthieu Constant[1] Isabelle Tellier[2] Denys Duchier[2]
Yoann Dupont[2] Anthony Sigogne[1] Sylvie Billot[2]
(1) Université Paris-Est, LIGM, CNRS, 5 bd Descartes, Champs-sur-Marne 77454 Marne-la-Vallée cedex 2
(2) LIFO, université d'Orléans, 6 rue Léonard de Vinci BP 6759, 45067 Orléans cedex 2
mconstan@univ-mlv.fr, isabelle.tellier@univ-orleans.fr, denys.duchier@univ-orleans.fr, yoann.dupont@etu.univ-orleans.fr, sigogne@univ-mlv.fr, sylvie.billot@univ-orleans.fr

Résumé. Dans cet article, nous synthétisons les résultats de plusieurs séries d'expériences réalisées à l'aide de CRF (Conditional Random Fields ou "champs markoviens conditionnels") linéaires pour apprendre à annoter des textes français à partir d'exemples, en exploitant diverses ressources linguistiques externes. Ces expériences ont porté sur l'étiquetage morphosyntaxique intégrant l'identification des unités polylexicales. Nous montrons que le modèle des CRF est capable d'intégrer des ressources lexicales riches en unités multi-mots de différentes manières et permet d'atteindre ainsi le meilleur taux de correction d'étiquetage actuel pour le français.

Abstract. In this paper, we synthesize different experiments using a linear CRF (Conditional Random Fields) to annotate French texts from examples, by exploiting external linguistic resources. These experiments especially dealt with part-of-speech tagging including multiword units identification. We show that CRF models allow to integrate, in different ways, large-coverage lexical resources including multiword units and reach state-of-the-art tagging results for French.

Mots-clés : Etiquetage morphosyntaxique, Modèle CRF, Ressources lexicales, Segmentation, Unités polylexicales.

Keywords: Part-of-speech tagging, CRF model, Lexical resources, Segmentation, Multiword units.

1 Introduction

Dans cet article, nous synthétisons les résultats de plusieurs séries d'expériences réalisées à l'aide de CRF (Conditional Random Fields ou "champs markoviens conditionnels" (Lafferty *et al.*, 2001; Tellier & Tommasi, 2011)) linéaires pour apprendre à annoter des textes français à partir d'exemples, en exploitant diverses ressources linguistiques externes. La tâche à laquelle nous nous sommes attachés est celle de la segmentation en unités lexicales des phrases d'un texte, couplée à celle de leur étiquetage en catégories morphosyntaxiques (ou "part of speech" en anglais).

Ces dernières années, l'étiquetage morphosyntaxique a atteint d'excellents niveaux de performance grâce à l'utilisation de modèles probabilistes discriminants comme les modèles de maximum d'entropie [MaxEnt] (Ratnaparkhi, 1996; Toutanova *et al.*, 2003), les séparateurs à vaste marge [SVM] (Giménez & Márquez., 2004) ou, déjà, les champs markoviens conditionnels [CRF] (Tsuruoka *et al.*, 2009). Il a par ailleurs été montré que le couplage de ces modèles avec des lexiques externes augmente encore la qualité de l'annotation, comme l'illustre (Denis & Sagot, 2009, 2010) pour MaxEnt. Néanmoins, les évaluations réalisées considèrent toujours en entrée un texte avec une segmentation lexicale parfaite, c'est-à-dire que les unités lexicales multi-mots, qui forment par définition des unités linguistiques, ont été parfaitement reconnues au préalable. Or cette tâche de segmentation est difficile car elle nécessite des ressources lexicales importantes. On notera que les systèmes tels que Macaon (Nasr *et al.*, 2010) et Unitex (Paumier, 2011) intègrent une analyse lexicale avec segmentation multi-mots ambiguë avant levée d'ambiguité par l'utilisation d'un modèle de Markov caché [HMM]. Dans cet article, nous proposons d'intégrer les deux tâches de segmentation et d'étiquetage dans un seul modèle CRF couplé à des ressources lexicales riches.

Le corpus d'apprentissage dont nous sommes partis provient du French Treebank (Abeillé *et al.*, 2003). Les ressources linguistiques externes utilisées sont de différentes natures. Nous avons ainsi exploité plusieurs dictionnaires : Lefff (Sagot, 2010) mais aussi DELA (Courtois, 2009; Courtois *et al.*, 1997), ainsi que des lexiques spécifiques comme Prolex (Piton *et al.*, 1999) et quelques autres incluant des noms d'organisation et des prénoms (Martineau *et al.*, 2009). Cet ensemble de dictionnaires est complété par une bibliothèque de grammaires locales qui reconnaissent différents types d'unités multi-mots (Constant & Watrin, 2008). Nous montrons que le modèle des CRF est capable d'intégrer de telles ressources de différentes manières et permet d'atteindre ainsi le meilleur taux actuel de correction pour la segmentation et l'étiquetage du français.

Dans la suite de cet article, nous commençons par présenter le modèle des CRF et le fonctionnement des bibliothèques logicielles que nous avons utilisées pour mener nos expériences. Nous décrivons ensuite le corpus d'apprentissage ainsi que la tâche que nous traitons, en détaillant les difficultés spécifiques que posent les unités multi-mots. Puis nous passons en revue les ressources à notre disposition et menons une réflexion méthodologique sur les différents moyens de les prendre en compte dans une chaîne de traitements qui fait appel à un CRF. La dernière partie est consacrée à la présentation des résultats de nos expériences. Ces travaux ont permis la mise au point de plusieurs segmenteurs-étiqueteurs qui sont librement disponibles.

2 Les CRF

2.1 Le modèle théorique

Les champs markoviens conditionnels ou CRF (Tellier & Tommasi, 2011) sont des modèles probabilistes discriminants introduits par (Lafferty *et al.*, 2001) pour l'annotation séquentielle. Ils ont été utilisés dans de nombreuses tâches de Traitement des Langues, où ils donnent d'excellents résultats (McCallum & Li, 2003; Sha & Pereira, 2003; Tsuruoka *et al.*, 2009; Tellier *et al.*, 2010).

Les CRF permettent d'associer à une observation x une annotation y en se basant sur un ensemble d'exemples étiquetés, c'est-à-dire un ensemble de couples (x, y). La plupart du temps (et ce sera le cas dans la suite de cet article), x est une *séquence d'unités* (ici, une suite d'unités lexicales) et y la *séquence des étiquettes correspondante* (ici, la suite de leurs catégories morphosyntaxiques, éventuellement enrichie pour coder la segmentation). Les CRF sont des modèles discriminants qui appartiennent à la famille des *modèles graphiques non dirigés*. Ils sont définis par X et Y, deux champs aléatoires décrivant respectivement chaque unité de l'observation x et son annotation y, et par un graphe $\mathcal{G} = (V, E)$ dont $V = X \cup Y$ est l'ensemble des nœuds (vertices) et $E \subseteq V \times V$ l'ensemble des

arcs (edges). Deux variables sont reliées dans le graphe si elles dépendent l'une de l'autre. Le graphe sur le champ Y des CRF linéaires, dessiné en Fig 1., traduit le fait que chaque étiquette est supposée dépendre de l'étiquette précédente et de la suivante et, implicitement, de la donnée x complète. Un dessin complet du graphe devrait ainsi également relier chaque variable Y_i à *chaque variable du champ X*, ce qu'on omet sur la figure pour la lisibilité.

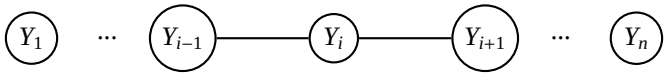

FIGURE 1 – graphe associé à un CRF linéaire

Dans un CRF, on a la relation suivante (Lafferty *et al.*, 2001) :

$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \exp\Big(\sum_k \lambda_k f_k(y_c, x, c) \Big) \quad avec$$

- C est l'ensemble des cliques (sous-graphes complètement connectés) de $\mathcal{G}$ sur Y : dans le cas du graphe de la Fig. 1, ces cliques sont constituées soit d'un nœud isolé, soit d'un couple de nœuds successifs.
- y_c l'ensemble des valeurs prises par les variables de Y sur la clique c pour un étiquetage y donné : ici, c'est donc soit la valeur d'une étiquette soit celles d'un couple d'étiquettes successives
- $Z(x)$ est un coefficient de normalisation, défini de telle sorte que la somme sur y de toutes les probabilités $p(y|x)$ pour une donnée x fixée soit égale à 1.
- Les fonctions f_k sont appelées *fonctions caractéristiques* (features) : elles sont définies à l'intérieur de chaque clique c et sont à valeurs réelles, mais souvent choisies pour donner un résultat binaire (0 ou 1). Elles doivent être founies au système par l'utilisateur. Par définition, la valeur de ces fonctions peut dépendre des étiquettes présentes dans une certaine clique c ainsi que de la valeur de *x n'importe où dans la donnée* (et pas uniquement aux indices correspondants à la clique c, ce qui donne beaucoup d'expressivité aux CRF).
- Les poids λ_k, à valeurs réelles, permettent d'accorder plus ou moins d'importance à chaque fonction f_k dont ils caractérisent le *pouvoir discriminant*. Ce sont les paramètres du modèle : l'enjeu de la phase d'apprentissage est de fixer leur valeur en cherchant à maximiser la log-vraisemblance sur un ensemble d'exemples déjà annotés (constituant le corpus d'apprentissage).

L'intérêt et l'efficacité des CRF proviennent de ce qu'ils prennent en compte des dépendances entre étiquettes reliées les unes aux autres dans le graphe. En cherchant le meilleur y, c'est-à-dire la meilleure *séquence d'étiquettes* associer à une donnée complète x, ils se comportent en général mieux qu'une série de classifications d'unités isolées. Mais cette prise en compte a un prix : la phase d'apprentissage d'un CRF peut être longue. Une fois cette phase réalisée, annoter une nouvelle séquence x de n mots en entrée revient alors à trouver le y qui maximise $p(y|x)$. L'espace théorique de recherche de ce meilleur étiquetage y est $|Y|^n$, où $|Y|$ est le nombre d'étiquettes distinctes possibles pour chaque nœud. Mais, grâce à des techniques de programmation dynamique, ce calcul peut être factorisé à l'intérieur des cliques et ramené à $K * n * |Y|^c$ où c est la taille de la plus grande clique ($c = 2$ pour les CRF linéaires) et K le nombre de fonctions caractéristiques. Une fois appris, l'étiqueteur est donc performant.

2.2 Les bibliothèques CRF++ et Wapiti

Notre objectif étant d'insérer des connaissances linguistiques dans un apprentissage réalisé à l'aide de CRF linéaires, il nous semble important de bien comprendre le fonctionnement concret des bibliothèques qui les implémentent. Plusieurs sont disponibles pour mettre en œuvre les CRF linéaires, notamment *crf.source.net* [1] de Sarawagi ou *Mallet* [2] de McCallum. Celles que nous avons utilisées sont *CRF++* [3] de Taku Kado et *Wapiti* [4] de Thomas Lavergne (Lavergne *et al.*, 2010), qui utilisent des moyens similaires pour instancier les fonctions caractéristiques qui entrent dans leur définition.

1. crf.sourceforge.net
2. http ://mallet.cs.umass.edu/
3. http ://crfpp.sourceforge.net/
4. http ://wapiti.limsi.fr

Corpus tabulaires. Les exemples d'apprentissage que requièrent ces bibliothèques sont des couples (x, y), où x est une séquence d'unités et y la séquence d'étiquettes correspondantes, de mêmes longueurs. Pour nous, une unité de x correspond à un "mot", mais elle peut être enrichie par d'autres propriétés, représentées par p attributs, du moment que ces derniers sont disponibles ou calculables aussi pour tout nouvel exemple x non étiqueté. Les attributs peuvent être des booléens (l'unité contient un chiffre, commence par une majuscule, est présente dans un lexique, etc.), des valeurs numériques (nombre de lettres, etc.) ou textuelles (valeur de l'unité ou de son préfixe ou suffixe de telle longueur, etc.). Une donnée étiquetée (x, y) de taille n se présente donc comme un tableau de n lignes et $p+1$ colonnes, où les p premières colonnes contiennent toutes les informations disponibles sur la donnée x et la dernière colonne les étiquettes y :

$$\begin{array}{cccccc} x_1^1 & x_1^2 & \cdots & x_1^p & y_1 \\ & & & \vdots & \\ x_i^1 & x_i^2 & \cdots & x_i^p & y_i \\ x_{i+1}^1 & x_{i+1}^2 & \cdots & x_{i+1}^p & y_{i+1} \\ & & & \vdots & \end{array}$$

Les exemples distincts sont séparés entre eux dans un même fichier par une ligne vide. Un corpus d'apprentissage est donc une suite de tels tableaux, tous de largeur $p + 1$, mais de hauteurs qui peuvent varier.

Patrons tabulaires. L'utilisateur des bibliothèques ne définit pas directement les fonctions caractéristiques du modèle ; il doit fournir des *patrons*. Il existe deux types de patrons correspondant aux deux tailles de clique possibles : les *unigrammes* pour les cliques de taille 1, et les *bigrammes* pour les cliques de taille 2.

Un patron unigramme est une sorte de carte perforée de même largeur $p + 1$ que nos tableaux, de hauteur quelconque sur les p premières colonnes mais ne pouvant capturer qu'une seule étiquette sur la colonne $p + 1$. Chaque position possible de cette carte sur un exemple définit une fonction caractéristique : celle qui renvoie la valeur 1 si la configuration de valeurs observée dans les perforations est satisfaite, 0 sinon. Les ronds dans le tableau précédent représentent les valeurs capturées par une telle carte, positionnée sur la ligne i d'une donnée. Chaque fonction caractéristique prend donc la forme d'une conjonction de critères booléens observée au moins une fois parmi les exemples et un patron en "génère" autant qu'il y a de positions où il peut s'appliquer dans le fichier d'exemples. Un patron permet de définir ainsi succinctement des milliers, voire des millions de fonctions caractéristiques. Un patron bigramme est similaire à un patron unigramme, mais on l'applique successivement à une position i, puis à la position suivante $i + 1$ et la fonction caractéristique obtenue est la conjonction de tous les critères rencontrés.

3 Corpus d'apprentissage pour la segmentation et l'étiquetage

3.1 Corpus FTB

Tout système d'annotation probabiliste supervisé requiert un corpus annoté de référence pour entraîner le modèle et ensuite l'évaluer. Pour notre tâche d'étiquetage morphosyntaxique intégrant la reconnaissance des unités multi-mots, il est donc nécessaire d'utiliser un corpus annoté en catégories grammaticales incluant l'annotation des unités polylexicales. Le corpus le plus complet en français est le corpus arboré de Paris 7 (Abeillé *et al.*, 2003), formé d'articles du journal *Le Monde* allant de 1989 à 1993. Il décrit la structure syntaxique des différentes phrases sous la forme d'arbres. Une unité de ce corpus peut être une ponctuation, un nombre, un mot simple ou une unité multi-mots. Au niveau morphosyntaxique, il existait initialement un jeu d'étiquettes de 14 catégories principales et de 34 sous-catégories. Pour notre tâche, nous utilisons un jeu d'étiquettes optimisé en 29 catégories pour l'analyse syntaxique (Crabbé & Candito, 2008) et réutilisé comme standard dans une expérience d'étiquetage morpho-syntaxique (Denis & Sagot, 2009). Les unités multi-mots codées sont de différents types : mots composés et entités nommées. Les mots composés comprennent des noms (*acquis sociaux*), des verbes (*faire face à*), des adverbes (*dans l'immédiat*), des prépositions (*en dehors de*). Il contient quelques types d'entités nommées : des noms d'organisation (*Société suisse de microélectronique et d'horlogerie*), des noms de famille (*Strauss-Kahn*), des noms de lieu (*Afrique du Sud, New York*).

Dans nos séries d'expériences, nous avons utilisé deux versions différentes du corpus : une version de 569 039 unités (au LIGM), une autre de 350 931 (au LIFO). Dans ces deux versions, nous n'avons repris que le niveau des feuilles, i.e. le niveau lexical. Nous en donnons un extrait ci-dessous :

```
Quant_à/P la/DET technique/NC ,/PONCT son/DET verdict/NC est/V implacable/ADJ ./PONCT
```

L'unité *Quant_à* est la fusion de deux mots simples (*Quant* et *à*), formant la préposition composée *quant_à*.

3.2 Unités lexicales multi-mots

Expressions multi-mots. Dans le consensus actuel du Traitement Automatique des Langues (TAL), les expressions multi-mots forment des unités linguistiques aux comportements lexicaux, syntaxiques et/ou sémantiques particuliers. Elles regroupent les expressions figées et semi-figées, les collocations, les entités nommées, les verbes à particule, les constructions à verbe support, les termes, etc. (Sag *et al.*, 2002). Leur identification est donc cruciale avant toute analyse sémantique. Elles apparaissent à différents niveaux de l'analyse linguistique : certaines forment des unités lexicales contigues à part entière (ex. *cordon bleu*, *San Francisco*, *par rapport à*), d'autres composent des constituants syntaxiques comme les phrases figées (*N0 prendre le taureau par les cornes* ; *N0 prendre N1 en compte*) ou les constructions à verbe support (*N0 donner un avertissement à N1* ; *N0 faire du bruit*).

Phénomènes traités. Dans cet article, nous ne traitons que les expressions multi-mots du niveau lexical, que nous appellerons dorénavant unités multi-mots ou polylexicales. Elles comportent les mots composés (noms, prépositions, adverbes, etc.), les entités nommées, les termes, les collocations nominales. Il existe une grande variété de phénomènes linguistiques rentrant dans cette catégorie et donc de nombreux critères d'identification. Les mots composés sont des séquences non compositionnelles de mots : ils présentent une opacité sémantique totale (*cordon bleu*, *tout à fait*) ou partielle (*vin blanc*), des contraintes syntaxiques et lexicales, etc. Il existe un continuum entre expressions figées et libres, ce qui rend leur identification encore plus difficile. Les collocations sont définies à partir de critères statistiques. Les entités nommées ont souvent une certaine compositionalité sémantique mais ont une syntaxe particulière : ex. *le 5 mars 2010* pour les dates, *Jacques Chirac* pour les noms de personnes.

Ressources. Les unités polylexicales peuvent être recensées dans des dictionnaires électroniques ou des grammaires locales. Les dictionnaires électroniques sont des listes qui associent des formes lexicales à des informations linguistiques comme les catégories grammaticales ou certains traits sémantiques (ex. *humain*, *concret*, etc.). Les grammaires locales (Gross, 1997; Silberztein, 2000) sont des réseaux récursifs de transitions décrits sous la forme de graphes d'automates finis. Chaque transition est étiquetée par un élément lexical (ex. *mange*), un masque lexical correspondant à un ensemble de formes lexicales encodées dans un dictionnaire (ex. <*manger*> symbolisant toutes les formes fléchies dont le lemme est *manger*) ou un élément non-terminal référant à un autre automate. Elles sont très utiles pour décrire de manière compacte des unités multi-mots acceptant des variations lexicales. Un système de transduction permet d'annoter les expressions décrites, comme la catégorie grammaticale ou l'analyse des composants internes pour les entités nommées par exemple (Martineau *et al.*, 2009).

Reconnaissance. La reconnaissance automatique des unités multi-mots est, la plupart du temps, réalisée à l'aide de ressources lexicales construites manuellement (ex. pour les expressions figées) ou apprises automatiquement (ex. collocations nominales). Par ailleurs, une grande partie des entités nommées, du fait de leur syntaxe particulière sont facilement décrites et reconnues à l'aide de grammaires locales (Friburger & Maurel, 2009; Martineau *et al.*, 2009), bien qu'il existe d'autres types d'approches telles que les systèmes statistiques (McCallum & Li, 2003) ou hybrides (Poibeau, 2009). L'identification de telles expressions est une tâche très difficile car les unités non décrites dans les ressources sont difficilement reconnaissables. Elle est d'autant plus difficile qu'elle dépend du contexte d'occurrence. En effet, une expression reconnue est souvent ambigue avec l'analyse en mots simples : par exemple, *il **en fait** une priorité* (mots simples) vs *j'ai **en fait** beaucoup travaillé* (mot composé). On observe parfois des chevauchements avec d'autres unités polylexicales comme dans la séquence *une pomme de terre cuite* où *pomme de terre* et *terre cuite* sont des mots composés. C'est pourquoi les outils existants de segmentation en unités multi-mots comme dans INTEX (Silberztein, 2000) ou SxPipe (Sagot & Boullier, 2008) produisent une segmentation ambiguë sous la forme d'automates finis acycliques pour éviter de prendre une décision définitive

trop hâtive. Cette analyse ambiguë peut alors être intégrée dans des traitements linguistiques tels que l'étiquetage morphosyntaxique (Nasr *et al.*, 2010; Paumier, 2011)) ou l'analyse syntaxique superficielle (Blanc *et al.*, 2007; Nasr *et al.*, 2010) et profonde (Sagot & Boullier, 2006).

3.3 Intégration d'un segmenteur et d'un étiqueteur

L'identification des unités multi-mots est similaire à une tâche de segmentation comme le chunking ou à la reconnaissance des entités nommées, qui identifient les limites de segments (chunks ou entités nommées) et les annotent. En effet, grâce à la représentation `IOB` [5] (Ramshaw & Marcus, 1995), segmenter un texte revient à annoter ses unités minimales. Pour combiner étiquetage morphosyntaxique et reconnaissance d'unités multi-mots, il suffit de concaténer les deux étiquetages en associant à chaque unité minimale une étiquette de la forme `X+B` ou `X+I`, où `X` est sa catégorie grammaticale et le suffixe indique si elle se trouve au début d'une unité multi-mots (`B`) ou dans une position "interne" (`I`). Le suffixe `O` est inutile car la fin d'un segment lexical correspond au début d'un autre (suffixe `B`) ou à une fin de phrase. Une telle procédure d'annotation détermine non seulement les limites des unités lexicales, mais aussi leur catégorie morphosyntaxique. Pour entraîner nos CRF, nous avons donc transformé le corpus d'apprentissage initial en isolant les unités composant les segments multi-mots et en les étiquetant conformément à cette nouvelle norme. L'exemple précédent est alors transformé en :

```
Quant/P+B à/P+I la/DET+B technique/NC+B ,/PONCT+B son/DET+B verdict/NC+B est/V+B
implacable/ADJ+B ./PONCT+B
```

Le jeu d'étiquettes initial est ainsi doublé, chaque étiquette se dédoublant en une variante `B` et une variante `I`. La reconnaissance des unités polylexicales dépendant fortement de la richesse de ressources lexicales utilisées, il s'agit maintenant de trouver les meilleures façons d'intégrer ce type d'informations dans nos CRF.

4 Exploitation d'une ressource externe

Dans cette section, nous commençons par présenter les différentes ressources que nous avons à notre disposition, et nous cherchons tous les moyens possibles de les prendre en compte dans un apprentissage avec des CRF.

4.1 Ressources

Même s'il existe de plus en plus d'études sur l'extraction automatique d'unités multi-mots, en particulier les collocations ou les termes (Daille, 1995; Dias, 2003; Seretan *et al.*, 2003), les ressources les plus riches et les plus précises ont été aquises manuellement. Pour notre étude, nous avons compilé diverses ressources lexicales sous la forme de dictionnaires morphosyntaxiques et de grammaires locales fortement lexicalisées. Nous avons utilisé notamment deux dictionnaires disponibles de mots simples et composés de la langue générale : DELA (Courtois, 2009; Courtois *et al.*, 1997) et Lefff (Sagot, 2010). Le DELA a été construit par une équipe de linguistes. Le Lefff a été automatiquement acquis et manuellement validé. Il résulte également de la fusion de différentes sources lexicales. En complément, nous disposons aussi de lexiques spécifiques comme Prolex (Piton *et al.*, 1999) composé de toponymes et d'autres incluant des noms d'organisation et des prénoms (Martineau *et al.*, 2009). Les nombres d'entrées de ces divers dictionnaires sont donnés dans le tableau 1.

Dictionnaire	#mots simples	#mots composés
DELA	690,619	272,226
Lefff	553,140	26,311
Prolex	25,190	97,925
Organisations	772	587
Prénoms	22,074	2,220

TABLE 1 – Dictionnaires morphosyntaxiques

5. I : Inside (intérieur du segment) ; O : Outside (hors du segment) ; B : Beginning (début du segment)

Cet ensemble de dictionnaires est complété par une bibliothèque de grammaires locales qui reconnaissent différents types d'unités multi-mots comme les entités nommées (dates, noms d'organisation, de personne et de lieu), prépositions locatives, déterminants numériques et nominaux. En pratique, nous avons utilisé une bibliothèque de 211 automates développée à partir de la bibliothèque en-ligne GraalWeb (Constant & Watrin, 2008).

4.2 Quelques statistiques préliminaires

Pour les expériences menées avec la variante du FTB la plus volumineuse, le corpus initial a été découpé en trois parties : 80% pour la phase d'entraînement (TRAIN), 10% pour le développement (DEV) et 10% pour le test. Cela nous a permis de faire quelques observations préalables.

Ainsi, dans le corpus FTB-DEV (avec étiquetage initial non transformé), nous avons observé qu'environ 97,4% des unités lexicales [6] sont présentes dans nos ressources lexicales (en particulier, 97% sont présentes dans les dictionnaires). Alors que 5% des unités sont inconnues (i.e. absentes du corpus d'apprentissage), 1,5% sont à fois inconnues et absentes des ressources lexicales, ce qui montre que 70% des unités inconnues sont couvertes par nos ressources. On observe également qu'environ 6% des unités sont multi-mots. En décomposant toutes les unités multi-mots du texte en unités minimales, on s'aperçoit qu'à peu près 15% d'entre elles sont incluses dans une unité multi-mots. Parmi les unités multi-mots codées dans le corpus FTB-DEV, 75,5% d'entre elles sont présentes dans nos ressources (87,5% en incluant le lexique du corpus d'entraînement). Ceci montre que 12,5% des unités multi-mots sont totalement inconnues et, par conséquent, seront sans doute très difficilement reconnaissables.

On observe, par ailleurs, que le corpus FTB ne couvre pas la reconnaissance de toutes les unités multi-mots. Tout d'abord, certains déterminants ou certaines entités nommées ne sont pas identifiés comme les déterminants nominaux, les dates, les noms de personne, les adresses postales. Par ailleurs, de nombreux noms composés sont manquants. Par exemple, après avoir appliqué nos ressources lexicales de manière non contextuelle (en excluant les grammaires locales reconnaissants des types d'entités nommées ou des déterminants nominaux non codés dans le FTB), nous avons manuellement observé sur le FTB-DEV qu'environ 30% des unités polylexicales de nos ressources "adaptées" ne sont pas prises en compte dans le corpus.

4.3 Méthodologie de prise en compte des ressources

Comment prendre en compte une ou plusieurs ressources lors d'une chaîne de traitements faisant appel à un apprentissage réalisé avec un CRF ? Dans le cadre de l'apprentissage de la ressource MElt$_{fr}$ (Denis & Sagot, 2009, 2010), les auteurs ont testé deux approches possibles :
– intégrer les propriétés des mots du lexique dans les fonctions caractéristiques du modèle d'apprentissage ;
– filtrer les étiquetages incompatibles avec les informations présentes dans la ressource.
Nous avons cherché toutes les façons possibles d'envisager cette intégration, ce qui nous a amené à en caractériser plus finement le mode opératoire, et à en trouver de nouvelles variantes. Nous les présentons ci-dessous, en discutant leurs intérêts et leurs limites. Elles peuvent s'organiser en deux familles principales, suivant que la ou les ressources disponibles sont mises à contribution comme des filtres *avant ou après* l'appel au CRF ou qu'elles sont utilisées *pendant la phase d'apprentissage*. L'approche "filtrage" requiert que les étiquettes qui figurent dans la ressource soient identiques à celles qui sont la cible de l'apprentissage, alors que ce n'est pas nécessairement le cas pour l'autre approche. Au cas où les conventions d'étiquetage ne sont pas les mêmes, une fonction de correspondance doit être préalablement appliquée.

Les ressources comme filtrage *a priori* ou *a posteriori* Les ressources peuvent être vues comme un moyen de contraindre, ou encore de *filtrer* les étiquetages possibles. Concrètement, ce filtrage peut opérer *avant* ou *après* l'appel au CRF. Le filtrage *a priori* consiste à définir l'espace de recherche des étiquetages possibles y d'une nouvelle chaîne x via un prétraitement fondé sur une ressource. Les analyseurs lexicaux actuels auxquels on soumet une phrase produisent en effet généralement un *dag* (graphe orienté acyclique) dont chaque chemin correspond à une séquence possible d'étiquettes. Les unités multi-mots peuvent être reconnues lors de cette étape, et figurer aussi dans le *dag*, comme cela a été évoqué section 3.2. Pour une phrase constituée de n unités minimales, il est évidemment plus facile et rapide de chercher le y qui maximise $p(y|x)$ (calculé suivant la formule des CRF) parmi

6. Les unités lexicales sont les unités autres que les nombres et les ponctuations.

l'ensemble des étiquetages du *dag* plutôt que sur l'espace de tous les $|Y|^n$ étiquetages possibles. Le filtrage est ainsi *a priori* mais l'apprentissage du CRF est néanmoins un pré-requis de la chaîne de traitements. Le filtrage *a posteriori*, lui, cherche non pas le meilleur étiquetage possible y d'une chaîne quelconque x mais les m meilleurs possibles (c'est une option généralement disponibles des bibliothèques CRF) et choisit le premier d'entre eux compatible avec la ressource. Les deux techniques donnent la même solution ; privilégier l'une ou l'autre dépend de la forme de la ressource. Leur intérêt est de garantir que dans la solution retenue, chaque mot reçoit une étiquette compatible avec ce que décrivent la ou les ressources consultées. Un filtrage peut d'ailleurs très bien se combiner avec une approche prenant en compte les ressources *pendant* la phase d'apprentissage.

Les ressources comme aide à l'apprentissage. D'après la section 2.2, quand nous faisons appel à une bibliothèque qui implémente les CRF linéaires, nous avons à notre disposition trois "leviers" d'action possibles :
– le choix des étiquettes et des propriétés des unités (les colonnes des données tabulaires)
– le choix des exemples (les lignes)
– le choix des fonctions caractéristiques (via les patrons), choix qui dépend fortement des précédents
Nous avons déjà vu qu'un choix pertinent d'étiquettes permettait de "coder" en quelque sorte les deux problèmes de la segmentation et de l'étiquetage simultanément. D'autres expériences ont montré l'intérêt de décomposer le jeu d'étiquettes en sous-étiquettes, notamment quand celles-ci sont trop nombreuses (Tellier *et al.*, 2010). Mais le problème auquel nous nous confrontons ici ne requiert pas un tel traitement, nous ne l'avons pas mis en œuvre.

Il est en revanche "naturel" d'insérer les informations des ressources en tant que propriétés des unités d'un exemple x, donc en jouant sur les colonnes $x_i^2, ..., x_i^p$. Plusieurs choix sont encore possibles pour cela, suivant qu'on se contente de concaténer les différentes étiquettes possibles d'une même unité pour en faire une seule colonne de nature textuelle, ou bien qu'on définisse autant de colonnes à valeur booléenne que d'étiquettes possibles dans l'ensemble de la ressource. Cela aura bien sûr des conséquences sur la définition des patrons qui génèrent les fonctions caractéristiques. Dans le cas des colonnes booléennes, la combinatoire des conjonctions possibles de plusieurs critères est explosive. Dans les deux cas, on peut soit garder les étiquettes des ressources telles quelles, soit les transformer pour qu'elles s'identifient à celles visées.

Enfin, il est aussi possible de considérer que chaque instance de couple (unité lexicale, étiquette) présent dans la ressource constitue à elle toute seule une "phrase" qu'on insère parmi les exemples étiquetés, en ajoutant de nouvelles lignes isolées dans le corpus d'apprentissage. Cela suppose bien sûr que les étiquettes qui figurent dans la ressource sont identiques à celles de l'étiquetage cible. L'idée sous-jacente de cette technique, très simple à appliquer, est que la présence dans une ressource équivaut à une occurrence attestée dans la langue, que l'on simule en l'insérant artificiellement dans le corpus d'apprentissage. Elle présente aussi toutefois quelques inconvénients :
– on introduit ainsi un biais sur les comptes d'occurrences puisque les différentes étiquettes possibles d'une unité donnent chacune lieu à un exemple, comme si elles étaient équiprobables. Il faut donc espérer que le reste de l'ensemble d'apprentissage soit suffisant pour compenser cette distorsion possible.
– en introduisant des "phrases" réduites à un mot, on va rendre inopérantes sur ces "phrases" particulières toutes les fonctions caractéristiques qui testent la valeur des unités ou des étiquettes voisins (et donc en particulier tous les bigrammes). Le poids de ces fonctions ne pourra être calculé que sur le reste des exemples.

5 Résultats des expériences

Les résultats présentés ici sont issus d'expériences menées en parallèle au LIFO (Orléans) et au LIGM (Paris-Est Marne-la-Vallée). Notons au préalable que les expériences ont été réalisées dans des environnements différents, sans coordination *a priori*, ce qui explique la difficulté à comparer précisément les résultats. Les expériences du LIFO ont été menées avec Wapiti [7] et évaluées par validation croisée en 10 parties : 9/10 pour l'apprentissage, 1/10 pour le test. Celles du LIGM ont utilisé CRF++ [8] et porté sur une variante du FTB plus volumineuse rendant plus coûteuse, mais aussi moins indispensable, une validation croisée : le corpus initial a alors été découpé en trois parties : 80% pour la phase d'entraînement (TRAIN), 10% pour le développement (DEV) et 10% pour le test.

Pour l'évaluation globale, nous avons précision = rappel = f-mesure. En effet, tous les mots ayant une unique étiquette, une erreur de précision sur une classe C1 correspond à une erreur de rappel sur une classe C2 et vice versa.

7. Ce programme a l'avantage d'opérer une sélection des fonctions caractéristiques *en cours d'apprentissage* grâce à une pénalisation L1.
8. L'algorithme de régularisation utilisé est L2 et le seuil de fréquence des traits a été fixé à 2.

Pour l'étiquetage avec segmentation, nous avons deux types d'évaluation : la f-mesure sur les unités minimales (LIFO) et sur les segments lexicaux (LIGM). Ceci explique les scores plus élevés pour LIFO sur cette tâche.

5.1 Evaluation de l'étiquetage avec segmentation parfaite

LIGM. Nous avons tout d'abord évalué l'étiquetage morphosyntaxique sur une segmentation multi-mots parfaite, au moyen d'un modèle CRF appris en utilisant des propriétés classiques des unités (forme lexicale, préfixes, suffixes, commence par une majuscule, etc.). Les expériences du LIGM ont porté sur deux méthodes d'intégration de la ressource lexicale externe décrite dans la section 4.1. La première méthode consiste à introduire, dans le fichier d'entraînement, une colonne supplémentaire (AC) représentant la concaténation des étiquettes trouvées dans la ressource pour l'unité courante. Nous obtenons alors un modèle LEX en utilisant tous les traits décrits dans la table 1(a). Nous notons STD le modèle incorporant les mêmes traits à l'exception de ceux issus de la ressource. La deuxième méthode consiste à procéder à un filtrage a priori de toutes les étiquettes absentes de la ressource pour chaque unité. Si l'unité est absente, toutes les étiquettes sont gardées. Les étiquettes des ressources ont été ajustées à celles du corpus pour le filtrage. Nous avons comparé les résultats avec d'autres outils d'étiquetage que nous avons tous entrainés sur le corpus FTB-TRAIN. Nous avons évalué TreeTagger (Schmid, 1994) basé sur des arbres de décision probabilistiques, SVMTool (Giménez & Márquez., 2004) basé sur les Séparateurs à Vastes Marges utilisant des traits indépendants de la langue, MElt (Denis & Sagot, 2009) basé sur un modèle MaxEnt incorporant en plus des traits dépendants de la langue issus de lexiques externes. Les lexiques utilisés pour entraîner et tester MElt intègrent toutes les ressources de la section 4.1 [9]. Les précisions obtenues sur le corpus FTB-TEST pour les différents systèmes sont données en pourcentage dans la table 1(b) avec un intervalle de confiance à 95% de +/-0,1.

(a) Types de traits

Traits internes unigrammes	
$w_0 = X$	$\&t_0 = T$
forme en minuscule de $w_0 = L$	$\&t_0 = T$
Préfixe de $w_0 = P$ avec $\|P\| < 5$	$\&t_0 = T$
Suffixe de $w_0 = S$ avec $\|S\| < 5$	$\&t_0 = T$
w_0 contient un tiret	$\&t_0 = T$
w_0 contient un chiffre	$\&t_0 = T$
w_0 commence par une majuscule	$\&t_0 = T$
w_0 est tout en majuscule	$\&t_0 = T$
w_0 commence par une majuscule et est en début de phrase	$\&t_0 = T$
classe d'ambiguité de w_0, $AC_0 = A$	$\&t_0 = T$
Traits contextuels unigrammes	
$w_i = X, i \in \{-2,-1,1,2\}$	$\&t_0 = T$
$w_i w_j = XY, (j,k) \in \{(-1,0),(0,1),(-1,1)\}$	$\&t_0 = T$
$AC_i = A, i \in \{-2,-1,1,2\}$	$\&t_0 = T$
Traits bigrammes	
$t_{-1} = T'$	$\&t_0 = T$

(b) Comparaison de systèmes d'étiquetage pour le français

	sans filtrage	avec filtrage
TreeTagger	96.4	-
SVMTool	97.2	-
MElt	97.6	-
CRF-STD	97.4	97.6
CRF-LEX	**97.7**	**97.7**

TABLE 2 – Résultats du LIGM avec segmentation parfaite

LIFO. Les expériences du LIFO ont porté sur une version du FTB moins volumineuse, en utilisant les traits unigrammes décrits dans la Table 3(a) sur une fenêtre $[-2..2]$ et les simples valeurs d'étiquettes pour les bigrammes. Les patrons bigrammes produisent en effet un très grand nombre de fonctions caractéristiques : cette restriction est destinée à limiter les calculs. La seule ressource à notre disposition était le Lefff. La première méthode utilisée pour le prendre en compte en cours d'apprentissage est de l'intégrer en tant que pourvoyeur de nouveaux exemples dans chaque fichier d'entraînement. Cette méthode augmente le temps d'apprentissage du simple au double voire triple selon les parties. La seconde méthode consiste à introduire des booléens en tant qu'attributs dans les colonnes des fichiers d'entraînement, chaque colonne représentant une étiquette possible dans le Lefff. Il a fallu alors générer par programme tous les patrons possibles qui combinent certains attributs entre eux. Cette méthode produit un grand nombre de fonctions caractéristiques mais Wapiti est capable de les gérer puisqu'il opère une sélection des fonctions caractéristiques les plus discriminantes *en cours d'apprentissage* (Lavergne *et al.*, 2010).

9. Nous avons regroupé ensemble tous les dictionnaires, ainsi que les unités reconnues lors de l'application des grammaires locales sur le corpus.

(a) Types de traits unigrammes

Valeur de l'unité
Commence par une majuscule
Est uniquement en majuscules
Est un chiffre
Est une ponctuation
3 dernières lettres

(b) Résultats

Sans lefff	96.5
Avec lefff (exemples)	96.6
Avec lefff (attributs booléens)	97.3

TABLE 3 – Résultats du LIFO avec segmentation parfaite

5.2 Evaluation de l'étiquetage avec identification des unités multi-mots

LIGM. Pour évaluer la tâche d'étiquetage intégrant la reconnaissance des unités multi-mots, nous avons entraîné trois modèles CRF sur le corpus FTB-TRAIN après avoir décomposé les unités multi-mots en séquences d'unités minimales étiquetées dans la représentation de type IOB (cf. section 3.3) : `STD`, `LEX` et `MWE`. Les deux premiers ont les mêmes types de traits que dans l'expérience précédente. Le modèle MWE est complété de traits issus de l'application non-contextuelle de nos ressources multi-mots sur le texte : une unité est associée à la catégorie grammaticale, la structure interne ou/et le trait sémantique de l'unité polylexicale reconnue à laquelle elle appartient, ainsi qu'à sa position relative dans l'unité (I, O ou B). Par exemple, le mot *de* dans le contexte du mot composé *eau de vie* présent dans nos ressources, sera associé à la catégorie grammaticale NC (nom), à la structure interne NPN (nom+préposition+nom) et à la position relative I (car il est en 2ème position). Ces trois systèmes ont été comparés avec SVMTool, entraîné sur le même corpus. Pour chaque segmenteur-étiqueteur appliqué sur le corpus TEST décomposé en unités minimales, nous avons calculé la f-mesure [10]. La précision et le rappel sont calculés par rapport aux segments lexicaux trouvés et non aux unités minimales simples. Les résultats sont synthétisés dans le tableau 3(a). La colonne `SEG` indique la f-mesure de la segmentation qui ne prend en compte que les limites des segments. La colonne `TAG` prend aussi en compte la catégorie grammaticale.

LIFO. Pour cette tâche, nous comparons les résultats obtenus (1) sans le Lefff, (2) avec le Lefff comme source d'exemples, (3) avec le Lefff comme source d'attributs booléens. Nos résultats évaluent la qualité de l'étiquetage des unités minimales avec les catégories intégrant B et I, et non celle de l'identification des unités multimots.

(a) LIGM (f-mesure sur les segments lexicaux)

	TAG	SEG
SVMTool	92.1	94.7
CRF-STD	93,7	95.8
CRF-LEX	93.9	95.9
CRF-MWE	**94.4**	**96.4**

(b) LIFO : méthodes d'intégration (f-mesure sur les unités minimales)

Sans lefff	94.5%
Avec lefff (exemples)	94.7%
Avec lefff (attributs)	95.2%

TABLE 4 – Apprentissage simultané étiquetage/segmentation

5.3 Description des segmenteurs-étiqueteurs proposés

Les diverses expériences décrites ci-dessus ont mené à la mise au point de segmenteurs-étiqueteurs qui sont librement disponibles. La chaîne de traitements de SEM [11] produite au LIFO a été écrite en Python. Le programme offre la possibilité d'exploiter ou non Lefff (sous forme d'attributs uniquement), en utilisant soit une segmentation rudimentaire écrite à la main (sans prise en compte de ressources externes), soit la segmentation acquise par le CRF. Le segmenteur-étiqueteur LGTagger [12] produit au LIGM est implanté en Java et comprend deux phases distinctes : (1) une analyse lexicale basée sur des ressources lexicales externes qui sert à filtrer les analyses (simples ou multi-mots) non décrites dans les ressources et qui produit un *dag* [13] ; (2) un décodeur qui détermine le

10. La formule de la f-mesure est la suivante : $f = \frac{2pr}{p+r}$ où p est la précision et r le rappel.
11. http ://www.univ-orleans.fr/lifo/Members/Isabelle.Tellier/SEM.html
12. http ://igm.univ-mlv.fr/~mconstan/research/software
13. Pour les mots simples inconnus de nos ressources, toutes les étiquettes possibles sont gardées comme candidates. Si l'analyseur n'a aucune ressource lexicale en entrée, il produit un *dag* représentant toutes les analyses possibles dans le jeu d'étiquettes.

chemin du *dag* le plus probable en fonction du modèle CRF appris. Il peut être exécuté avec ou sans segmentation multi-mots. Les ressources lexicales (pour le calcul des propriétés des fonctions caractéristiques et pour l'analyse lexicale) lui sont passées en paramètres. Les programmes d'Unitex (Paumier, 2011) sont utilisés pour l'application des ressources : consultation des dictionnaires et application des grammaires locales.

6 Conclusion

Dans cet article, nous avons montré que les tâches de segmentation et d'étiquetage sont intimement liées et qu'il est naturel de les traiter simultanément. L'écart entre la performance de l'étiquetage avec ou sans segmentation est de 2 à 4 points suivant la mesure utilisée : cela mesure le "coût" d'une bonne segmentation. Par ailleurs, nous avons montré l'intérêt certain d'intégrer des ressources lexicales dans un CRF, en particulier les ressources d'unités polylexicales utiles pour la segmentation. Nous voyons aussi qu'à ce niveau de performance, il est extrêmement difficile de gagner quelques dixièmes de points, même en mettant en jeu des ressources riches et variées.

Cet article a aussi été l'occasion d'une réflexion méthodologique poussée sur les différents moyens d'intégrer une ressource linguistique externe dans une chaîne de traitements faisant appel à un CRF. Une bonne partie de cette réflexion est d'ailleurs transposable à l'utilisation d'autres techniques d'apprentissage automatique. Les CRF, en intégrant fonctions caractéristiques locales et combinaison statistique globale, apparaissent comme un modèle particulièrement bien adapté à l'hybridation entre ressources symboliques et modèles statistiques. Grâce à cette intégration, il a été possible de produire en peu de temps des segmenteurs-étiqueteurs très performants.

Références

ABEILLÉ A., CLÉMENT L. & TOUSSENEL F. (2003). Building a treebank for french. In A. ABEILLÉ, Ed., *Treebanks*. Dordrecht : Kluwer.

BLANC O., CONSTANT M. & WATRIN P. (2007). Segmentation in super-chunks with a finite-state approach. In *Proceedings of the 6th Workshop on Finite-State Methods and Natural Language Processing (FSMNLP'07)*, p. 62 – 73.

CONSTANT M. & WATRIN P. (2008). Networking multiword units. In *Proceedings of the 6th International Conference on Natural Language Processing (GoTAL'08)*, number 5221 in Lecture Notes in Artificial Intelligence, p. 120 – 125 : Springer-Verlag.

COURTOIS B. (2009). Un système de dictionnaires électroniques pour les mots simples du français. *Langue Française*, **87**, 1941 – 1947.

COURTOIS B., GARRIGUES M., GROSS G., GROSS M., JUNG R., MATHIEU-COLAS M., MONCEAUX A., PONCET-MONTANGE A., SILBERZTEIN M. & VIVÉS R. (1997). *Dictionnaire électronique DELAC : les mots composés binaires*. Rapport interne 56, University Paris 7, LADL.

CRABBÉ B. & CANDITO M. H. (2008). Expériences d'analyse syntaxique statistique du français. In *Actes de TALN 2008 (Traitement automatique des langues naturelles)*, Avignon.

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.

DENIS P. & 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 2009)*.

DENIS P. & SAGOT B. (2010). Exploitation d'une ressource lexicale pour la construction d'un étiqueteur morphosyntaxique état-de-l'art du francais. In *actes de TALN 2010*.

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 (ACL 2003)*, p. 41–49.

FRIBURGER N. & MAUREL D. (2009). Finite-state transducer cascade to extract named entities in texts. *Theoretical Computer Science*, **313**, 94–104.

GIMÉNEZ J. & MÁRQUEZ. 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)*.

Gross M. (1997). The construction of local grammars. In D. J. Lipcoll, D. H. Lawrie & A. H. Sameh, Eds., *Finite-State Language Processing*, p. 329–352. Cambridge, Mass. : The MIT Press.

Lafferty J., McCallum A. & 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 2001)*, p. 282–289.

Lavergne T., Cappé O. & Yvon F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 504–513 : Association for Computational Linguistics.

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

McCallum A. & Li W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL*.

Nasr A., Béchet F. & Rey J. F. (2010). Macaon : Une chaîne linguistique pour le traitement de graphes de mots. In *Traitement Automatique des Langues Naturelles - session de démonstrations*, Montréal.

Paumier S. (2011). Unitex 2.1 - user manual. http ://igm.univ-mlv.fr/˜unitex.

Piton O., Maurel D. & 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)*, p. 233–237.

Poibeau T. (2009). Boosting Robustness of a Named Entity Recognizer. *International Journal of Semantic Computing*, **3**(1), 1–14.

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

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 1996)*, p. 133 – 142.

Sag I. A., Baldwin T., Bond F., Copestake A. A. & 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)*, p. 1–15, London, UK : Springer-Verlag.

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)*.

Sagot B. & Boullier P. (2006). Deep non-probabilistic parsing of large corpora. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*.

Sagot B. & Boullier P. (2008). Sxpipe 2 : architecture pour le traitement pré-syntaxique de corpus bruts. *Traitement Automatique des Langues*, **49**(2), 155–188.

Schmid H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, p. 252 – 259.

Seretan V., Nerima L. & Wehrli E. (2003). Extraction of multi-word collocations using syntactic bigram composition. In *Proceedings of the Fourth International Conference on Recent Advances in NLP (RANLP-2003)*, p. 424–431, Borovets, Bulgaria.

Sha F. & Pereira F. (2003). Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*, p. 213 – 220.

Silberztein M. (2000). Intex : an fst toolbox. *Theoretical Computer Science*, **231**(1), 33–46.

Tellier I., Eshkol I., Taalab S. & Prost J. P. (2010). Pos-tagging for oral texts with crf and category decomposition. *Research in Computing Science*, **46**, 79–90. Special issue "Natural Language Processing and its Applications".

Tellier I. & Tommasi M. (2011). Champs Markoviens Conditionnels pour l'extraction d'information. In Eric Gaussier & François Yvon, Eds., *Modèles probabilistes pour l'accès à l'information textuelle*. Hermès.

Toutanova K., Klein D., Manning C. D. & Singer Y. Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, p. 252 – 259.

Tsuruoka Y., Tsujii J. & 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 2009)*, p. 790–798.