Un point sur les outils du LPL pour l'analyse syntaxique du français

Stéphane Rauzy, Philippe Blache
LPL, CNRS & Université de Provence
Aix-en-Provence
{stephane.rauzy; philippe.blache}@lpl-aix.fr

Abstract

Nous présentons ici les différents modules et ressources développés au Laboratoire Parole et Langage pour l'analyse syntaxique du français. Ces outils sont basés sur des approches symboliques ou stochastiques, selon les caractéristiques de la tâche à effectuer. La chaîne de traitement est composée d'un segmenteur par règles et d'un lexique couvrant du français qui alimentent l'entrée d'un étiqueteur morphosyntaxique probabiliste. Deux analyseurs de surface, l'un symbolique et l'autre stochastique, sont ensuite présentés. Un analyseur stochastique profond, récemment développé, est aussi proposé.

1 Introduction

Les outils d'analyse syntaxique sont aujourd'hui confrontés à la nécessité de traiter des données variées, non canoniques, comme la langue parlée. Il s'agit d'un problème bien entendu complexe, d'une part à cause de la variabilité des constructions mais surtout par le fait qu'elles sont fréquemment partielles, pour différentes raisons. Il est nécessaire pour traiter ce type de données de tirer parti de toutes les informations disponibles, provenant de différents domaines comme la prosodie, mais également de la structure du discours, ou de la pragmatique. Les approches probabilistes constituent clairement une réponse adaptée à ce type de problème. Il est cependant nécessaire de disposer pour leur développement de corpus d'entraînement contenant toutes ces informations. La constitution de ce type de ressources est aujourd'hui un enjeu essentiel, pour toute les langues, et en particulier pour le français qui fait preuve d'un certain retard en la matière, faute d'efforts coordonnés dans ce sens.

Nous proposons dans cet article de faire le point sur les outils et ressources développés au LPL dans cette perspective. Précisons que nous avons porté dans un premier temps nos efforts sur la question de la segmentation, quel que soit son niveau : il s'agit de rechercher des frontières d'unités, ou, plus généralement, de positionner des balises dans un flux de symboles. Nous avons pour cela mis au point une technique efficace (cf. (Blache and Rauzy, 2006)) qui a été généralisée et appliquée à différents problèmes comme la segmentation en phrases ou pseudo-phrases (unité pertinente pour l'oral), le repérage de constituants prosodiques, le chunking ou l'analyse syntaxique.

Concrètement, nous avons, comme il est d'usage dans ce type d'opération, commencé par traiter le niveau lexical et morpho-syntaxique en développant d'une part un lexique morphologique et d'autre part les outils de traitement de premier niveau (segmenteur, étiqueteur). Pour ce qui concerne le niveau syntaxique, nous avons développé deux types d'analyseurs, reposant sur des techniques symboliques et numériques. Ces deux approches sont en effet complémentaires. Les premières, outre l'intérêt théorique qu'elles présentent pour la validation des formalismes et des grammaires, peuvent être mises en oeuvre directement. Elles sont donc utiles pour la création de ressources qui permettront l'entraînement des outils probabilistes. Concrètement, pour ce qui concerne l'analyse syntaxique, nous avons créé un treebank de 1500 arbres en deux étapes : production des arbres par l'analyseur symbolique, puis correction manuelle. Le même procédé en deux étapes est utilisé pour adapter progressivement nos outils au traitement de la langue parlée.

Ce document propose une présentation rapide des principales caractéristiques des ressources et outils développés ou en cours de développement au LPL.

2 Segmentation et lexique

Cette étape a pour objectif de découper le texte en entrée en une séquence de segments (*tokens*), puis d'associer à chacun de ces tokens la liste des catégories morphosyntaxiques lui correspondant. Cette tâche, si elle ne présente aucune difficulté particulière, est pourtant d'une importance cruciale. Les erreurs de segmentation ainsi que les approximations dues à une mauvaise qualité du lexique se répercutent en effet tout au long de la chaîne de traitement.

La phase de segmentation permet de répérer les frontières séparant les tokens et d'identifier les entités nécessitant un traitement spécial (nombres, dates, heures, noms propres, sigles, ...). Nous utilisons dans notre chaîne de traitement un segmenteur par règles qui associe à chaque entité une règle (sous forme d'une expression régulière) et une liste d'exceptions. En cas de conflit, par exemple lorsque deux entités identifiées se chevauchent, la solution d'empan maximum est préférée. Une fois la séquence de tokens formés, l'information syntaxique associée à chaque token est obtenue par accès au lexique.

Le lexique DicoLPL (Vanrullen et al., 2005) est un lexique couvrant du français ($440\,000$ formes) régulièrement corrigé et mis à jour, voir par exemple (Rauzy and Blache, 2007). La clé d'entrée du lexique se présente sous la forme du couple graphie-catégorie morphosyntaxique. Pour chaque entrée, le lexique propose une fréquence lexicale, extraite d'un corpus de textes écrits d'environ 150 million de mots préalablement étiqueté, voir (Vanrullen et al., 2005) pour plus de détails.

L'information sur les fréquences lexicales permet notamment de préciser, pour une graphie ambiguë, la répartition observée entre les catégories qui lui sont associées. Par exemple, la graphie `dans` est rencontrée dans le corpus $1\,056\,924$ fois sous forme de préposition contre 195 fois sous forme de nom commun, alors que la graphie `envers` se distribue plus uniformément ($5\,174$ pour la préposition, $2\,123$ pour le nom commun). la prise en compte de cette information, de nature extra-syntaxique, améliore considérablement la qualité de l'étiquetage morphosyntaxique. Nous montrons dans (Blache and Rauzy, 2008) que cette information permet à elle seule d'atteindre un score de désambiguisation morphosyntaxique de 0.89 (F-Mesure).

3 Etiquetage et désambiguisation

La procédure de désambiguisation consiste à associer à chacun des tokens de l'énoncé une catégorie morphosyntaxique unique. Nous utilisons ici le modèle des patrons (Blache and Rauzy, 2006; Blache and Rauzy, 2007), un modèle de Markov caché (HMM) plus performant que les modèles de type *N-grammes*. Pour les N-grammes, les états de l'automate sont identifiés par des séquences de catégories de taille identique $N-1$. Le modèle des patrons relâche cette contrainte en acceptant des états identifiés par des séquences de longueur variable, voir par exemple (Ron et al., 1996). Cette caractéristique permet en pratique d'extraire du corpus d'apprentissage un ensemble d'états, les *patrons* du modèle, qui capture de façon optimale l'information contenue dans le corpus.

Chaque patron du modèle est caractérisé par :

- son identifiant, c'est-à-dire la séquence de catégories qui le constitue, par exemple la suite de catégories `Det Adj Noun`.
- un vecteur donnant la probabilité de transition vers chaque catégorie du modèle, conditionnée par l'identifiant du patron, par exemple $\mathtt{p}(\mathtt{Verb} \mid \mathtt{Det\ Adj\ Noun})$[1] pour la probabilité de transition vers la catégorie `Verb`.
- un vecteur donnant pour chaque transition possible, le nouvel état atteint par le système (le *patron cible*). Par exemple, si au système dans l'état `Det Adj Noun` est proposé la catégorie `Verb`, le système évolue vers l'état `Noun Verb`.

Par construction, le patron cible associé à une transition ne dépend pas du chemin suivi dans l'espace des états du système. Cette propriété permet en pratique d'utiliser des algorithmes de programmation dynamique très efficaces, comme l'algorithme de Viterbi, pour calculer la solution la plus probable. Ici, on ne suppose pas que les données vérifient effectivement cette hypothèse d'indépendance. Le modèle des patrons est vu comme une approximation du modèle statistique décrivant réellement les données, une approximation qui est d'autant plus fidèle que le corpus d'apprentissage possède une taille importante. Des phénomènes comme les dépendances non bornées par exemple, ne seront pas capturés par le modèle

[1] $\mathtt{p}(\mathtt{Verb} \mid \mathtt{Det\ Adj\ Noun})$ est la probabilité de `Verb` conditionnée par le contexte gauche `Det Adj Noun`.

des patrons.

Pour la tâche de désambiguisation, le problème consiste à trouver pour la séquence de tokens, chaque token proposant sa liste de catégories potentiellement réalisables, la séquence de catégories de probabilité maximale. Ceci équivaut à trouver le chemin optimal suivi dans le treillis des états du système compte tenu des choix proposés par la séquence de tokens en entrée. Cette solution est obtenue par application de l'algorithme de Viterbi.

Le modèle est entrainé sur le corpus Grace/Multitag (Paroubek and Rajman, 2000), un échantillon d'environ $700\,000$ mots annoté morphosyntaxiquement selon le jeu de traits Multext (Ide and Véronis, 1994). L'information morphosyntaxique disponible a été groupée de manière ad-hoc en 44 catégories distinctes (2 types de catégories pour les ponctuations, 1 pour les interjections, 2 pour les adjectifs, 2 pour les conjonctions, 1 pour les déterminants, 3 pour les noms, 8 pour les auxiliaires, 4 pour les verbes, 5 pour les prépositions, 3 pour les adverbes et 11 pour les pronoms). Les informations comme les traits d'accords en genre, nombre et personne ou le temps des verbes ne sont pas exploitées dans la version actuelle de l'étiqueteur.

Le modèle des patrons complet est composé de $3\,053$ patrons de taille variable (le plus grand contexte dans la liste des patrons est composé d'une séquence de 6 catégories). L'évaluation du modèle est réalisé en comparant l'étiquetage optimal obtenu à partir du modèle (par application de l'algorithme de Viterbi) et la référence. Pour le jeu de catégories sélectionnées (les 44 catégories mentionnées précédemment), la qualité de l'étiqueteur atteint un score de 0.94 (F-mesure), voir (Blache and Rauzy, 2008) pour plus de détails.

4 Analyse syntaxique superficielle

Nous avons proposé deux analyseurs superficiels dans le cadre de la campagne d'évaluation Passage 2007[2] (de la Clergerie et al., 2008) qui est une continuation de la campagne Easy (Paroubek et al., 2006). Ces campagnes ont permis une évaluation comparative de plusieurs analyseurs syntaxiques du français en s'appuyant sur un format d'annotation ad-hoc, le guide d'annotation PEAS (Gendner et al., 2003), proposant des unités syntaxiques plates (i.e. sans constituants emboîtés). Deux évaluations distinctes étaient proposées durant ces campagnes, l'une portant sur la tâche de formation et d'identification des groupes syntaxiques, l'autre consistant à établir les relations de dépendance entre les groupes obtenus. Nos analyseurs superficiels traitent du premier problème classique de chunking : il s'agit de repérer les types et frontières des constituants de niveau 1.

4.1 L'analyseur symbolique ShP1

Il s'agit d'un analyseur déterministe. Il repose sur les Grammaires de Propriétés avec une stratégie de coin gauche. La grammaire utilisée est complète en ce sens qu'elle peut être utilisée indifféremment pour une analyse profonde ou superficielle (Balfourier et al., 2005). La particularité de ShP1 est de s'appuyer sur un sous-ensemble de contraintes de la grammaire (en particulier les propriétés de linéarité et de constituance) pour identifier les coins gauches. La stratégie consiste à repérer à partir des coins gauches la frontière droite du chunk sur la base des autres propriétés. Cette heuristique est très efficace et permet à l'analyseur de bénéficier d'une grande rapidité (moins de 4 minutes pour traiter 1M de mots). Dans le cadre de la campagne Passage 2007, cet analyseur a obtenu une F-Mesure de $91.57\,\%$.

Cet analyseur a été adapté pour les besoin des campagnes d'évaluation Easy et Passage au format de sortie requis. Au delà des chunks, il produit cependant des arbres syntaxiques complets. Il a ainsi été utilisé pour produire l'étape initiale du treebank du LPL : 1500 arbres, générés par cet analyseur, ont ensuite été corrigés manuellement ; formant ainsi une ressource syntaxique complémentaire à celles existantes.

4.2 L'analyseur stochastique StP1

L'analyseur stochastique StP1, comme notre étiqueteur, est basé sur le modèle des patrons. La grammaire Easy compte six constituants (les groupes Easy `GN`, `GP`, `NV`, `GA`, `PV` et `GR`) non emboîtables. Dans notre représentation, les catégories non-terminales sont annotées par leur catégories ouvrante et fermante associées, par exemple `<GN>` et `</GN>` pour le groupe `GN`. Pour chaque catégorie terminale (c'est-à-dire les 44 catégories terminales introduites section 3), on définit six catégories mentionnant le groupe dominant la catégorie et une catégorie

[2] L'action Passage : http ://atoll.inria.fr/passage/

supplémentaire lorsque la catégorie est isolée. Par exemple GN/Noun représente la catégorie Noun dans un groupe GN, GP/Noun représente la catégorie Noun dans un groupe GP, ./Pct la catégorie Pct isolée. Au total, si 320 catégories sont définies, beaucoup d'entre elles ne sont pas présentes en pratique, par exemple GN/Verb (il n'y a pas de verbe dans un groupe nominal).

La phase d'apprentissage est effectuée sur le gold standard Easy, un corpus annoté en constituants d'environ 100 000 mots qui a servi de référence pour la campagne d'évaluation Easy (Paroubek et al., 2006). Le corpus ne fournit pas les étiquettes des tokens composant les énoncés. Une phase préalable d'étiquetage (en utilisant l'étiqueteur présenté section 3) a donc été nécessaire pour produire notre échantillon d'apprentissage. L'échantillon est ensuite encodé selon la définition des catégories décrites ci-dessus.

Le module d'apprentissage nous permet d'extraire de cet échantillon 1340 patrons de taille variable identifiés par des séquences de catégories terminales ou de catégories ouvrantes et fermantes associées aux groupes. Pour la tâche qui nous concerne ici, c'est-à-dire insérer au sein d'une séquence de catégories observées (les catégories terminales) les catégories non-observées (les catégories ouvrantes et fermantes), une étape supplémentaire est requise. Pour chaque patron du modèle dont l'identifiant se termine par une catégorie observée (*patron observé*), et pour chaque catégorie observée, est établi la liste des transitions possibles vers cette catégorie, avec ou sans insertion. Pour chaque item de cette liste est calculé la probabilité de transition, le patron cible et le vecteur de catégories non-observées insérées pour réaliser cette transition. Par exemple, pour le patron observé GN/Det GN/Noun et pour la catégorie observée NV/Verb, il existe une transition vers le patron cible <NV> NV/Verb avec insertion des catégories </GN><NV>. A la fin de la procédure, nous obtenons un modèle constitué de 1080 patrons observés.

L'analyseur stochastique StP1 prend en entrée un texte étiqueté et désambiguisé ou une liste de catégories associée à chaque token de l'énoncé (la sortie de la phase segmentation plus accès au lexique). Pour chaque énoncé, l'algorithme de Viterbi permet d'insérer les groupes Easy maximisant la probabilité de l'énoncé. Dans le cadre de la campagne Passage 2007, StP1 a obtenu une F-Mesure de 93.03 %.

5 Analyse syntaxique profonde

Comme indiqué dans l'introduction, nous avons porté notre attention sur la question de la segmentation et l'identification des unités. Il s'agit en effet d'un premier niveau d'information essentiel dans l'analyse des interactions entre les différents domaines de l'analyse linguistique. Concrètement, nous avons décidé dans un premier temps de développer un analyseur produisant des arbres, sans nous préoccuper de la question des relations grammaticales. Dans l'état actuel des ressources disponibles, l'identification des fonctions est en cours d'intégration, sur la base des informations disponibles dans le French Treebank (FTB, (Abeillé et al., 2001; Abeillé et al., 2003)). Dans un second temps, le développement d'un treebank adapté nous permettra de proposer l'identification de toutes les relations.

5.1 Un analyseur stochastique profond

Nous avons récemment développé un analyseur stochastique profond basé sur le modèle des patrons. Nous présentons ici les modifications qui ont été apportées au formalisme originel afin de traiter les structures arborescentes. Une description plus détaillée de la méthode sera proposée ultérieurement dans (Rauzy, en préparation).

La première modification concerne le calcul de la probabilité d'une séquence de catégories. Pour probabiliser les structures d'arbres, nous faisons l'hypothèse que la probabilité d'une catégorie conditionnée par un syntagme ne dépend pas des constituants du syntagme. Ainsi, on considère que la probabilité conditionnelle p(Verb|<NP>Det Adj Noun</NP>) est équivalente à p(Verb|NP), ceci quelques soient les constituants du syntagme NP. C'est une hypothèse forte dont la validité dépend de la définition effective des syntagmes de la grammaire. Néanmoins, une fois vérifiée, cette hypothèse permet de définir un procédure de réduction très efficace pour le calcul de la probabilité d'une structure arborescente.

Nous illustrons par un exemple table 1 le calcul de la probabilité pas à pas. Le patron courant est initialisé sur le patron début de phrase <SENT>. La catégorie ouvrante <NP> est ajoutée au système, avec une probabilité de transition p(<NP>|<SENT>) = 0.4, le système évolue

t	back	patron	cat	proba
0	-	<SENT>	<NP>	0.4
1	-	<SENT><NP>	Det	0.7
2	-	<SENT><NP>Det	Noun	0.8
3	0	<NP>Det Noun	</NP>	0.3
4	-	<SENT>NP	<VP>	0.6
5	-	<SENT>NP<VP>	Verb	0.5
6	-	NP<VP>Verb	<NP>	0.3
7	-	<VP>Verb<NP>	Det	0.7
8	-	Verb<NP>Det	Noun	0.7
9	6	<NP>Det Noun	</NP>	0.3
10	3	<VP>Verb NP	</VP>	0.4
11	-	<SENT>NP VP	Pct	0.9
12	-	<SENT>NP VP Pct	</SENT>	1.0

TAB. 1 – Le calcul pas à pas de la probabilité d'une séquence arborée. Colonne 1, l'indice de l'étape ; colonne 2, la position cible du backtracking lorsque une réduction est réalisée ; colonne 3, l'état courant du système ; colonne 4, la catégorie ajoutée ; colonne 5, la probabilité de transition du patron courant vers la catégorie ajoutée.

vers le patron cible `<SENT><NP>` associé à cette transition. Les catégories `Det` et `Noun` sont ajoutée à l'étape 1 et 2, la probabilité cumulée de la séquence étant le produit des probabilités de transition à chaque étape. A l'étape 3 apparaît le premier phénomène de réduction. La catégorie fermante `</NP>` est ajoutée, avec une probabilité de transition `p(</NP>|<NP>Det Noun) = 0.3`. Une procédure de *backtracking* permet de remonter jusqu'à l'étape 0 pour former le patron cible `<SENT>NP` associé à la réduction. Après cette réduction, le système est dans l'état `<SENT>NP`, la catégorie `<VP>` est alors ajoutée avec une probabilité de transition `p(<VP>|<SENT>NP) = 0.6`. La probabilité de la séquence arborée est obtenue en poursuivant l'évolution du système jusqu'à la fin de la phrase.

Dans le cadre de ce schéma, l'identifiant des patrons du modèle est une séquence formée de catégories terminales, de catégories ouvrantes, et de catégories non-terminales. En revanche, les catégories fermantes sont interdites. Pour chaque patron, les probabilités de transition vers les categories ouvrantes, fermantes, terminales et non-terminales sont calculées par le module d'apprentissage à partir d'un corpus arborée. Le patron cible associé à chaque transition est aussi mémorisé, sauf pour les transitions vers les catégories fermantes pour lesquelles le patron cible est obtenu dynamiquement par backtracking sur l'historique des états du système.

La tâche d'analyse consiste à insérer au sein d'une séquence de catégories observées (les catégories terminales) les catégories non-observées (les catégories ouvrantes et fermantes). On ne conservera donc dans le modèle que les patrons observés (i.e. dont l'identifiant se termine par une catégorie observée). Pour chaque patron observé, et pour chaque catégorie observée ou chaque syntagme formé, est établi la liste des transitions possibles vers cette catégorie ou ce syntagme, avec ou sans insertion. Pour chaque item de cette liste est calculé la probabilité de transition, le patron cible et le vecteur de catégories ouvrantes insérées pour réaliser cette transition. Par exemple, pour le patron observé `NP<VP>Verb` et pour la catégorie observée `Det`, il existe une transition vers le patron cible `Verb<NP>Det` avec insertion de la catégorie `<NP>`. De plus, le patron observé mémorise la probabilité de transition vers chaque catégorie fermante. Nous illustrons table 2 un exemple de construction d'un arbre à partir de la séquence des catégories observées.

t	back	patron	insert	cat
0	-	<SENT>	<NP>	Det
1	-	<SENT><NP>Det	-	Noun
2	0	<NP>Det Noun	</NP>	-
3	-	<SENT>NP	<VP>	Verb
4	-	NP<VP>Verb	<NP>	Det
5	-	Verb<NP>Det	-	Noun
6	4	<NP>Det Noun	</NP>	-
7	3	<VP>Verb NP	</VP>	-
8	-	<SENT>NP VP	-	Pct
9	-	<SENT>NP VP Pct	</SENT>	-

TAB. 2 – L'évolution pas à pas de la construction de l'arbre à partir de la séquence des catégories observées. Colonne 1, l'indice de l'étape ; colonne 2, la position cible du backtracking lorsque une réduction est réalisée ; colonne 3, l'état courant du système ; colonne 4, le vecteur de catégories insérées ; colonne 5, la catégorie observée ajoutée.

Du fait du schéma de réduction, la transition vers le patron cible lors de la procédure de backtracking dépend de l'historique des états du système. Les algorithmes du type Viterbi sont alors inopérants pour trouver la structure d'arbre maximisant la probabilité, pour un séquence de catégories observées en entrée. Nous avons ici opté pour un algorithme du type *beam search* qui va élaguer dynamiquement l'arbre des solutions compte tenu de la probabilité de chaque solution au cours du temps. Une procédure de rattrapage est implémentée lorsque aucune solution n'est retenue

par l'algorithme, ce qui se produit par exemple lorsque la solution à atteindre a été supprimée de liste des solutions conservées lors d'une étape précédente du fait de sa faible probabilité.

6 Conclusion et perspectives

Le développement d'outils d'analyse syntaxique robustes, permettant le traitement de corpus volumineux avec un niveau d'information adéquat repose sur la possibilité de disposer de techniques complémentaires, permettant en particulier le développement de ressources utiles à l'apprentissage et l'entraînement d'analyseurs stochastiques. Nous sommes donc engagés dans des développements multiples : ressources lexicales, corpus arborés, outils de pré-traitement, analyseurs symboliques et stochastiques. Les travaux en cours au LPL portent plus particulièrement sur trois aspects : mise au point d'un corpus arboré avec les relations grammaticales, finalisation d'un analyseur stochastique profond et adaptation des outils au traitement de l'oral. Les outils et ressources décrits dans ce document sont disponibles pour utilisation à des fins de recherche. Certains sont d'ores et déjà accessibles via le CRDO (http ://www.crdo.fr).

References

A. Abeillé, C. Clément, A. Kinyon, and F. Toussenel. 2001. Un corpus français arboré : quelques interrogations. In *Actes de Traitement Automatique des Langues Naturelles*, volume 1, pages 33–42, Tours, France, 2-5 juillet.

A. Abeillé, L. Clément, and F. Toussenel. 2003. Building a treebank for french. In A. Abeillé, editor, *Treebanks*, Kluwer, Dordrecht.

J.-M. Balfourier, P. Blache, M.-L. Guénot, and T. Vanrullen. 2005. Comparaison de trois analyseurs symboliques pour une tâche d'annotation syntaxique. In *Actes de Traitement Automatique des Langues Naturelles*, volume 2, pages 41–48, Dourdan, France, 6-10 juin.

P. Blache and S. Rauzy. 2006. Mécanismes de contrôle pour l'analyse en grammaires de propriétés. In *Actes de Traitement Automatique des Langues Naturelles*, pages 415–424, Leuven, Belgique, 10-13 avril.

P. Blache and S. Rauzy. 2007. Le moteur de prédiction de mots de la plateforme de communication alternative. *Traitement Automatique des Langues*, 48(2) :47–70.

P. Blache and S. Rauzy. 2008. Influence de la qualité de l'étiquetage sur le chunking : une corrélation dépendant de la taille des chunks. In *Actes de Traitement Automatique des Langues Naturelles*, pages 290–299, Avignon, France, 9-13 juin.

E. de la Clergerie, C. Ayache, G. de Chalandar, G. Francopoulo, C. Gardent, and P. Paroubek. 2008. Large scale production of syntactic annotations for french. In *Proceedings of the international workshop on Automated Syntactic Annotations for Interoperable Language Resources*, Hong-Kong.

V. Gendner, G. Illouz, M. Jardino, L. Monceaux, P. Paroubek, I. Robba, and A. Vilnat. 2003. PEAS, the first instantiation of a comparative framework for evaluating parsers of french. In *Research Notes of EACL 2003*, Budapest, Hongrie, Avril.

Nancy Ide and Jean Véronis. 1994. MULTEXT : Multilingual text tools and corpora. In *Proceedings of the 15th. International Conference on Computational Linguistics (*COLING 94*)*, volume I, pages 588–592, Kyoto, Japan.

P. Paroubek and M. Rajman. 2000. Multitag, une ressource linguistique produit du paradigme d'évaluation. In *Actes de Traitement Automatique des Langues Naturelles*, pages 297–306, Lausanne, Suisse, 16-18 octobre.

P. Paroubek, I. Robba, A. Vilnat, and C. Ayache. 2006. Data annotations and measures in EASY the evaluation campaign for parsers in french. In *Proceedings of the 5th international Conference on Language Resources and Evaluation*, pages 314–320, Genoa, Italy, May.

S. Rauzy and P. Blache. 2007. Un lexique syntaxique des verbes du français : VfrLPL. In *Rapport de recherche RAU-3055*, Laboratoire Parole et Langage.

D. Ron, Y. Singer, and N. Tishby. 1996. The power of amnesia : Learning probabilistic automata with variable memory length. *Machine Learning*, 25 :117–149.

T. Vanrullen, P. Blache, C. Portes, S. Rauzy, J.-F. Maeyhieux, M.-L. Guénot, J.-M. Balfourier, and E. Bellengier. 2005. Une plateforme pour l'acquisition, la maintenance et la validation de ressources lexicales. In *Actes de Traitement Automatique des Langues Naturelles*, volume 1, pages 511–516, Dourdan, France, 6-10 juin.