Création d'un nouveau treebank à partir de quatrièmes de couverture

Philippe Blache, Grégoire Montcheuil, Stéphane Rauzy, Marie-Laure Guénot

Création d'un nouveau treebank à partir de quatrièmes de couverture

Philippe Blache[1] Grégoire Montcheuil[2] Stéphane Rauzy[1] Marie-Laure Guénot[2]
(1) Aix Marseille Université, CNRS, Laboratoire Parole et Langage UMR 7309
(2) Equipex ORTOLANG, CNRS
{prenom.nom}@lpl-aix.fr

Résumé. Nous présentons ici 4-couv, un nouveau corpus arboré d'environ 3 500 phrases, constitué d'un ensemble de quatrièmes de couverture, étiqueté et analysé automatiquement puis corrigé et validé à la main. Il répond à des besoins spécifiques pour des projets de linguistique expérimentale, et vise à rester compatible avec les autres treebanks existants pour le français. Nous présentons ici le corpus lui-même ainsi que les outils utilisés pour les différentes étapes de son élaboration : choix des textes, étiquetage, parsing, correction manuelle.

Abstract.

Creation of a new treebank with backcovers

We introduce 4-couv, a treebank of approximatively 3 500 trees, built from a set of literacy backcovers. It has been automatically tagged and parsed, then manually corrected and validated. It was developed in the perspective of linguistic expriment projects, and aims to be compatible with other standard treebanks for french. We present in the following the corpus itself, then the tools we used or developed for the different stages of its elaboration : texts' selection, tagging, parsing, and manual correction.

Mots-clés : Corpus arboré, Étiquetage automatique, Analyse syntaxique automatique, Parsing stochastique, Conventions d'annotation, Outils d'annotation, Linguistique expérimentale.

Keywords: Treebank, Tagging, Parsing, Stochastic parsing, Annotation scheme, Annotation tools, Experimental linguistics.

1 Introduction

Les treebanks constituent une ressource indispensable non seulement pour la description de la syntaxe d'une langue, mais également pour l'entraînement ou la validation des systèmes d'analyse automatique. Pour le français, les premières ressources véritablement disponibles restent bien entendu le *French Treebank* (Abeillé & Crabbé, 2013) et toutes ses déclinaisons. Il existe également d'autres ressources, notamment la partie française du *Universal Dependencies Treebank* [1]. Cependant, si l'on élargit le champs d'application au cadre de la **linguistique expérimentale**, il devient nécessaire de disposer de ressources de haut niveau pouvant être utilisées dans des expériences avec des sujets humains. Il est dans ce cas crucial de prendre en compte des éléments tels que la nature des textes, leur style, leur lisibilité. Le type d'expérience typique pour l'étude du traitement du langage par l'homme consiste à analyser les données associées à la lecture (mouvement oculaire, électro-encéphalographie). Or, l'intérêt de disposer d'un treebank pour faire passer ce type d'expérience est très important : il devient en effet possible d'établir des modèles prédictifs (par exemple de difficulté) à partir d'informations syntaxiques. Toutefois la plupart des études conduites à ce jour ne prennent en compte que le niveau morphosyntaxique (Demberg & Keller, 2008). Ces expériences ont consisté à faire lire des textes à un certain nombre de sujets. Une expérience similaire a été conduite pour le français (Rauzy & Blache, 2012) en utilisant des extraits du FTB. À cette occasion, nous avons pu constater un biais important dans l'acquisition des données : la **nature des textes** eux-mêmes. Il s'agit en effet d'articles du *Monde*, anciens, et dont l'intérêt est souvent très limité. Un effet de lecture superficielle suscité par le manque d'intérêt est alors important, entraînant une chute d'attention importante, en même temps qu'un déficit de compréhension.

Nous avons donc décidé, à la fois pour enrichir le patrimoine existant, mais également dans la perspective de pouvoir les

1. https://code.google.com/p/uni-dep-tb/

utiliser dans un environnement expérimental, de constituer un nouveau treebank, sur la base de textes courts, sémantiquement consistants (en d'autres termes, auto-suffisants pour leur interprétation), et suscitant l'intérêt de façon à maintenir l'attention pendant la lecture. Nous avons pour cela choisi de constituer un corpus de "quatrièmes de couverture", permettant de respecter les contraintes indiquées : il s'agit du **corpus 4-Couv**, en cours de développement et dont une première livraison pourra être effectuée fin 2015. Ce corpus est constitué d'un ensemble de textes provenant de différents éditeurs (Pocket, Gallimard) ayant donné leur accord pour un usage à fins de recherche. Nous avons ainsi pu récupérer environ 8 000 textes. Un premier corpus de 500 textes a été constitué, représentant environ 3 500 phrases, formant la première livraison du treebank 4-Couv.

Nous proposons dans cet article de décrire la méthodologie et les outils mis au point pour la constitution de 4-Couv. Au-delà des problèmes classiques soulevés pour la constitution de ce type de treebank (analyse automatique, correction manuelle) s'ajoute l'étape de **sélection de textes** parmi un ensemble volumineux : nous avons pour cela développé un système d'aide à la sélection, décrit dans la première partie. Il s'agit d'un outil permettant d'évaluer (manuellement) certains critères tout en effectuant au passage un certain nombre de vérifications ou corrections (segmentations, mots inconnus, etc.). Il est ensuite nécessaire de traiter les problèmes posés par l'analyse syntaxique. Nous décrivons ainsi dans la seconde partie la question du **jeu d'étiquettes et des annotations syntaxiques** ; afin d'assurer une certaine interopérabilité, nous proposons pour cela de rester dans le cadre proposé par le FTB, en introduisant quelques modifications mineures. Nous avons ainsi développé un **analyseur syntaxique**, décrit dans la 3ème partie, entraîné sur le FTB ainsi modifié, et qui nous permet de produire les arbres d'entrée du treebank. Le résultat est enfin corrigé manuellement. Nous décrivons dans la dernière partie deux systèmes que nous avons développés dans cette perspective : un **système d'aide à la correction morphosyntaxique** et un **éditeur d'arbres syntaxiques**.

2 Constitution du corpus et outil d'aide à la sélection

2.1 Description des textes

Comme nous l'avons dit précédemment, nous construisons le corpus à partir des descriptions qui se trouvent sur la quatrième de couverture des livres. Une petite étude statistique réalisée sur 1 000 textes pris au hasard avant sélection nous a confirmé que ce sont des textes assez courts : 136 742 tokens [2] pour 6 838 phrases, 80% des textes ont entre 80 et 200 tokens et entre 4 et 10 phrases ; la taille moyenne des phrases est de 20 tokens (80% des phrases ont moins de 30 tokens et moins de 10% en ont plus de 40). Les textes correspondent généralement à (a) un extrait du livre, (b) un résumé ou synopsis de l'histoire, (c) la genèse du texte, (d) un commentaire à propos de l'œuvre, ou encore (e) une combinaison de deux ou trois de ces éléments. Chacun de ces textes courts est sémantiquement autonome et, élément crucial pour notre corpus, est censé entretenir l'intérêt à la lecture, en minimisant autant que possible la chute de l'attention et de la recherche de compréhension.

2.2 Outil d'aide à la sélection

Afin de choisir les textes les plus pertinents pour le corpus, nous avons mis au point un système d'aide à la sélection basé sur des fichiers HTML constituant de véritables petits wikis autonomes présentant une dizaine de textes à évaluer. Cette stratégie de fichiers HTML autonomes permet de répartir facilement le travail de relecture à diverses personnes, sans la nécessité d'installer un logiciel particulier (les fichiers étant utilisables directement par la plupart des navigateurs web modernes [3]), ni de se connecter à un serveur central (ce qui permet un travail hors-ligne). Nous nous sommes basés pour cela sur l'outil TiddlyWiki [4] qui nous fournit le squelette des fichiers de wikis autonomes que nous avons configuré pour nos besoins. Nous avons ensuite utilisé un script Perl pour "remplir" chaque fichier avec les informations des quatrièmes de couverture d'une dizaine de livres.

Comme le montre la figure 1, pour chaque texte un "tiddler" [5] est créé présentant les sections suivantes : (a) les métadonnés du livre (auteur, titre, éditeur, ISBN,...), (b) le texte dans sa présentation initiale [6], (c) la découpe en phrases du

2. Les tokens incluent les mots et les signes de ponctuation.
3. Seule l'installation d'un petit complément étant parfois nécessaire pour la sauvegarde des fichiers.
4. `http://classic.tiddlywiki.com/` (version 2.8.1)
5. L'unité d'information de base dans TiddlyWiki qui, dans notre cas, correspond à un onglet.
6. En réalité seules quelques informations typographiques sont préservées : paragraphe, italiques, gras,...

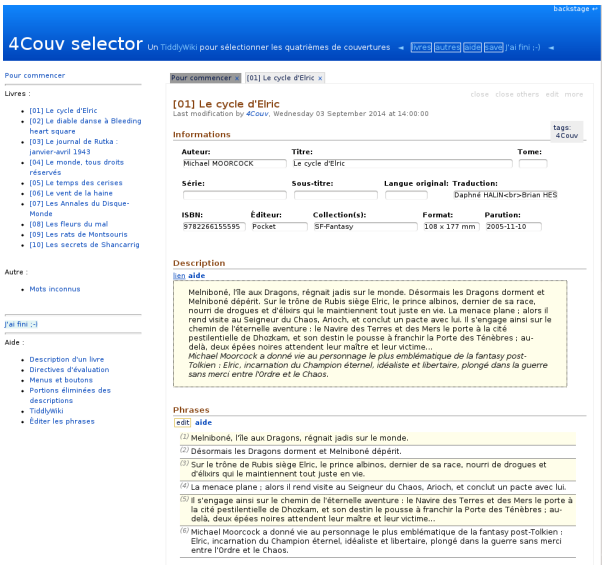

FIGURE 1 – Outil de sélection : vue générale

texte, (d) un cadre d'évaluation, (e) une liste des mots non-reconnus par le POS tagger. La section d'évaluation se compose essentiellement de cases à cocher et de champs à sélectionner pour simplifier celle-ci. D'autre part, grâce à la syntaxe wiki, il est relativement aisé de corriger d'éventuelles erreurs dans la découpe en phrases (celles-ci sont les lignes d'une table) ou d'introduire des limites de sections dans la composition de texte (en ajoutant des lignes blanches). Enfin, au cas où ce serait nécessaire, les champs de la section méta-données constituent un formulaire qui permet des rectifications.

3 Les annotations syntaxiques

3.1 L'étiquetage lexical

Pour l'annotation des unités syntaxiques minimales on se base sur un lexique [7] qui associe à chaque forme une étiquette lexicale (partie du discours) et un vecteur de traits de sous-catégorisation. Le découpage en tokens est maximal, dans le sens où l'on découpera en unités lexicales distinctes même les formes très contraintes dès lors qu'elles obéissent aux règles de construction syntaxique standard ; p.ex. on étiquettera séparément les constituants d'expressions semi-figées telles que *il était une fois* ou bien *mettre à nu*, mais pas ceux de formes telles que *d'autant plus* ou *tant mieux* parce que celles-ci ne répondent plus à des contraintes syntaxiques générales.

À chaque catégorie lexicale correspond un jeu de traits spécifique, bien que de nombreux traits se retrouvent sur plusieurs catégories (typiquement le genre, le nombre, la personne). Les catégories ainsi que les jeux de traits utilisés sont somme toute assez standard, compatibles avec la plupart des corpus étiquetés automatiquement, et permettent d'indiquer un ensemble d'informations lexicales, morphologiques, syntaxiques ou parfois sémantiques qui auront une incidence sur la construction syntaxique des unités aux niveaux supérieurs, p.ex. le nombre d'un déterminant, les compléments attendus par un verbe ou le cas d'un pronom clitique.

Nous n'avons pas de constituants lexicaux discontinus, ni ne conservons d'ambiguité dans l'étiquetage (i.e., tous les éléments reçoivent une catégorie lexicale, dont les traits de sous-catégorisation peuvent être sous-spécifiés le cas échéant). On ne modifie pas la catégorie des unités qui changent de paradigme (*une tarte maison, il est très zen*).

7. MarsaLex, `https://www.ortolang.fr/#/market/item/02f75cf8-8fcd-4305-a1ec-b34d516e716c`

3.2 L'annotation syntaxique

Les unités lexicales et syntaxiques entretiennent des relations syntagmatiques que l'on représente sous forme d'arbres. Ici aussi, afin de veiller à la compatibilité avec les autres treebanks existants, nous observons les contraintes de forme suivantes :

– On n'introduit pas de catégories vides dans les arbres (p.ex. dans le cas d'une construction elliptique) : chaque noeud est instancié par une unité lexicale ou syntagmatique.

– On fait une distinction entre niveau lexical et niveau syntagmatique, qui fait que l'on pourra avoir des syntagmes unaires, p.ex. *Simone* sera le constituant unique d'un NP dans (1).

(1) *Simone m'en donne trois*

– L'annotation ne comporte pas de constituants discontinus. Il s'agit d'une contrainte forte qui s'applique sur les choix linguistiques d'analyse que l'on peut faire, p.ex. pour des constructions présentant des discontinuités dans la structure syntagmatique, comme on trouve dans (1) ou (2), pour lesquelles notre liberté d'analyse se trouve limitée par cette obligation de forme.

(2) *Ce film, Paul et moi on a adoré*

– Pour annoter les syntagmes, on veille à n'attribuer que des étiquettes correspondant à des constructions proprement syntagmatiques (typiquement syntagmes nominaux, verbaux, adjectivaux, etc.) ; cela a pour conséquence notable que notre annotation des constructions coordonnées, phénomène canoniquement problématique, est différente de celle utilisée par le FTB et ses déclinaisons.

– On utilise le même type d'annotation des fonctions syntaxiques que celui introduit pour le FTB [8].

Notre approche de l'annotation syntaxique est guidée par les usages (*corpus driven*, cf. p.ex. Lacheret *et al.* (2014) parmi les projets récents), c'est-à-dire que la correction manuelle des arbres obtenus automatiquement peut mener, rétroactivement, à des modifications du fonctionnement du parseur (qui réapprend sur les sorties corrigées) afin d'en améliorer les résultats de manière dynamique.

4 Génération automatique de treebank

Le treebank est généré à partir de l'analyseur stochastique du LPL (Rauzy & Blache, 2009). La chaine de traitement suit un schéma classique. Dans un premier temps, le texte brut est segmenté en tokens par un segmenteur à base de règles. Un lexique permet ensuite d'associer à la forme de chaque token la distribution des catégories morphosyntaxiques correspondantes. Le processus de désambiguisation est réalisé par un étiqueteur stochastique utilisant la technologie HMM (Rabiner, 1989) pour identifier la séquence de catégories morphosyntaxiques la plus probable. Dans un dernier temps, un analyseur stochastique permet de générer les structures d'arbre aptes à décrire chaque énoncé et de sélectionner la structure d'arbre la plus probable décrivant l'énoncé. Cette chaine de traitement est implémentée dans *MarsaTag* (Rauzy *et al.*, 2014).

Le modèle probabiliste pour la phase d'étiquetage est entrainé sur le corpus GraceLPL, une version du corpus Grace/Multitag (Paroubek & Rajman, 2000) contenant 700 000 tokens, que nous corrigeons et enrichissons régulièrement. L'information morphosyntaxique est dans notre modèle organisée sous la forme de 48 étiquettes distinctes (version 2013). Sur ce jeu d'étiquettes, l'évaluation de notre étiqueteur atteint un score de 0.974 (F-mesure).

Le modèle probabiliste pour l'analyseur est obtenu à partir du corpus FTLPL (Blache & Rauzy, 2012), une version du MFT (Schluter & van Genabith, 2007) extrait du FTB (Abeillé *et al.*, 2001). Le corpus FTLPL compte actuellement 1 500 phrases validées (soit environ 26 000 tokens), pour lesquelles la structure en constituants (*AP, NP, VP*,...) et leurs fonctions syntaxiques (*SUBJ, OBJ, ATR*, ...) sont disponibles. L'information syntaxique est retenue dans le modèle sous forme de 36 constituants distincts (en tenant compte des fonctions différentes associées aux constituants). L'algorithme de génération des arbres et de la sélection de l'arbre le plus probable s'inscrit dans l'approche *Augmented Transition Network* (Woods, 1970). L'évaluation complète de notre analyseur stochastique reste à venir. Sur les 1 500 phrases du corpus d'apprentissage, l'analyseur associe la même analyse que la référence pour 500 phrases environ (même structure d'arbre et mêmes étiquettes de constituants et de fonction), et la même structure d'arbre (mais avec des étiquettes différentes) pour 900 d'entre elles.

8. Ce type d'annotation est malheureusement moins précis et plus *ad hoc* que le système qui avait p.ex. été utilisé dans Gendner *et al.* (2009) où informations constructionnelles et fonctionnelles étaient mentionnées indépendamment.

5 Les outils de correction utilisés

La correction des annotations automatiques est réalisée en deux étapes. La première concerne la **correction de l'étiquetage morphosyntaxique** ; l'interface de correction est illustrée figure 2. La seconde étape consiste à **réviser les arbres syntagmatiques** produits à l'aide de l'outil *MarsaTag* (Rauzy *et al.*, 2014).

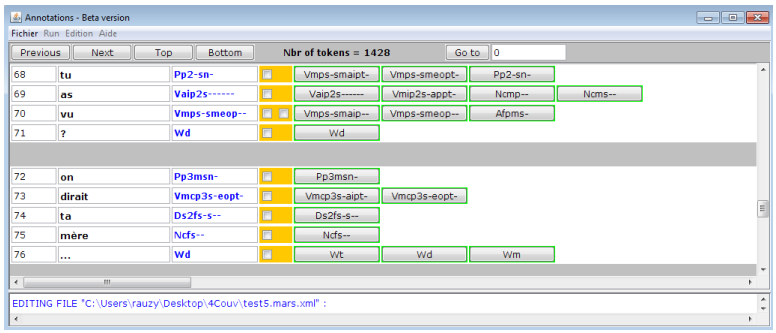

FIGURE 2 – L'interface utilisée pour corriger l'étiquetage. *Le texte est présenté horizontalement, une ligne par token. Chaque ligne contient la forme, la solution retenue par l'étiqueteur (deuxième colonne), et la liste des étiquettes possibles associées à chaque forme. Pour corriger une étiquette, l'annotateur clique sur l'étiquette désirée dans la liste ou saisira les traits de l'étiquette si celle-ci n'est pas proposée.*

Rares sont les éditeurs d'arbres syntaxiques adaptés au formalisme de constituants. Parmi les applications "installées" nous pouvons citer WordFreak (Morton & LaCivita, 2003) utilisé dans de nombreux projets d'annotation ou encore TrED 2.0 (Pajas & Štěpánek, 2008) qui nous a semblé l'un des plus complets pour ses grandes possibilités de personnalisation. En outre, ces dernières années ont vu l'émergence de nouvelles plateformes d'annotation linguistique totalement "en ligne", c'est-à-dire basées sur une architecture client-serveur, permettant souvent une annotation intuitive et rapide de textes (brat (Stenetorp *et al.*, 2012), TextAE [9]) et intégrant parfois une gestion du processus d'annotation et des différents "rôles" comme ceux d'annotateur, de curateur ou de chef de projet (GATE Teamware (Bontcheva *et al.*, 2013), WebAnno (Yimam *et al.*, 2013)). Cependant, si ces derniers outils sont assez bien adaptés au formalisme de dépendance, qui ne nécessite que deux "étages" d'annotation : les tokens et les liens de dépendance entre ceux-ci, ils ne sont pas vraiment adaptés au formalisme de constituants, où de nombreux "étages" de syntagmes peuvent se superposer. Pour conserver la simplicité d'un outil accessible avec n'importe quel navigateur web, nous avons donc créé une librairie JavaScript d'édition d'arbre syntaxique qui peut-être intégrée à une simple page html (figure 3) — et pour laquelle nous travaillons également à son intégration dans une plateforme d'annotation. Nous avons utilisé la librairie d3.js [10] qui permet de générer des images SVG dynamiques, auxquelles il est possible d'appliquer des styles CSS pour personnaliser l'affichage. L'édition de l'arbre se fait à l'aide de déplacement de nœuds (*drag & drop*) et de fonctions d'édition accessibles via le menu contextuel.

6 Conclusion

Cet article a une double vocation : présenter un ensemble d'outils pour le treebanking et présenter un nouveau type de treebank, adapté à l'acquisition de données comportementales, physiologiques et cérébrales pour l'étude du traitement du langage.

Du point de vue des **outils** (disponibles via la plateforme de ressources SLDR@ORTOLANG [11]), nous disposons désormais d'un environnement complet pour la constitution de treebank, permettant la sélection de textes et leur pré-traitement, l'analyse syntaxique automatique ainsi que deux outils d'aide à la correction : éditeur morpho-syntaxique et éditeur d'arbres.

Le **treebank** ainsi construit est composé d'un ensemble de textes courts, consistants du point de vue discursif (*i.e.* formant une unité discursive complète) et adaptés à l'expérimentation (par exemple pour l'acquisition de mouvements oculaires).

9. http://textae.pubannotation.org/
10. http://d3js.org/
11. http://sldr.org/

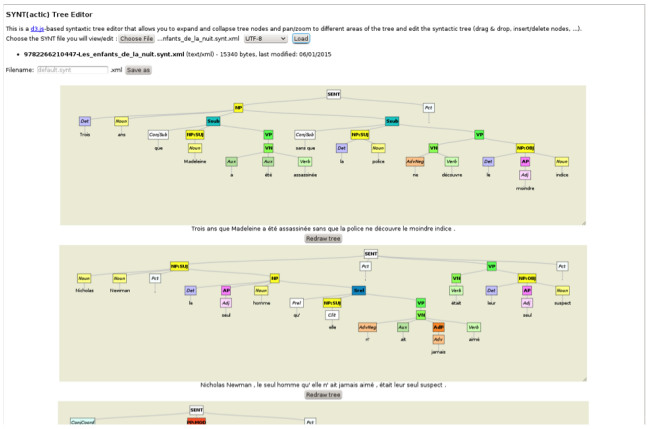

FIGURE 3 – Éditeur d'arbre : vue générale

Les outils présentés ici sont actuellement utilisés pour la constitution d'un premier treebank de 500 textes (3 500 arbres) qui sera opérationnel fin 2015, et lui-même également distribué via SLDR@ORTOLANG à des fins de recherche scientifique.

Nous sommes actuellement engagés dans le développement d'un treebank comparable en mandarin avec la collaboration de Hong-Kong Polytechnic University), dont une partie est formée de quatrièmes de couvertures d'ouvrages existants dans les deux langues.

Références

ABEILLÉ A., CLÉMENT C., KINYON A. & TOUSSENEL F. (2001). Un corpus français arboré : quelques interrogations. In *Actes de Traitement Automatique des Langues Naturelles*, volume 1, p. 33–42, Tours, France.

ABEILLÉ A. & CRABBÉ B. (2013). Vers un treebank du français parlé. In *Actes de TALN*.

BLACHE P. & RAUZY S. (2012). Enrichissement du FTB : un treebank hybride constituants/propriétés. In *Actes de TALN*.

BONTCHEVA K., CUNNINGHAM H., ROBERTS I., ROBERTS A., TABLAN V., ASWANI N. & GORRELL G. (2013). Gate teamware : a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, **47**(4), 1007–1029.

DEMBERG V. & KELLER F. (2008). Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, **109**(2), 193–210.

GENDNER V., VILNAT A., MONCEAUX L., PAROUBEK P., ROBBA I., FRANCOPOULO G. & GUÉNOT M.-L. (2009). *Les annotation syntaxiques de référence PEAS*. Rapport interne, version 2.2.

HERNANDEZ N. & BOUDIN F. (2013). Construction automatique d'un large corpus libre annoté morpho-syntaxiquement en français. In *Actes de la conférence TALN-RECITAL 2013*.

LACHERET A., KAHANE S., BELIAO J., DISTER A., GERDES K., GOLDMAN J.-P., OBIN N., PIETRANDREA P. & TCHOBANOV A. (2014). Rhapsodie : Un treebank annoté pour l'étude de l'interface syntaxe-prosodie en français parlé. In *4e congrès mondial de linguistique française*, volume 8, p. 2675–2689.

MORTON T. & LACIVITA J. (2003). Wordfreak : An open tool for linguistic annotation. In *Proceedings of NAACL-Demonstrations '03*, p. 17–18, Stroudsburg, PA, USA : Association for Computational Linguistics.

PAJAS P. & ŠTĚPÁNEK J. (2008). Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, p. 673–680, Manchester, UK : Coling 2008 Organizing Committee.

PAROUBEK P. & RAJMAN M. (2000). Multitag, une ressource linguistique produit du paradigme d'évaluation. In *Actes de Traitement Automatique des Langues Naturelles*, p. 297–306, Lausanne, Suisse.

RABINER L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257–286.

RAUZY S. & BLACHE P. (2009). Un point sur les outils du lpl pour l'analyse syntaxique du français. In *Actes du workshop ATALA 'Quels analyseurs syntaxiques pour le français ?'*, p. 1–6, Paris, France.

RAUZY S. & BLACHE P. (2012). Robustness and processing difficulty models. a pilot study for eye-tracking data on the french treebank. In *Proceedings of Workshop on Eye-tracking and Natural Language Processing at The 24th International Conference on Computational Linguistics (COLING)*.

RAUZY S., MONTCHEUIL G. & BLACHE P. (2014). MarsaTag, a tagger for French written texts and speech transcriptions. In *Second Asia Pacific Corpus Linguistics Conference*, Hong Kong.

SCHLUTER N. & VAN GENABITH J. (2007). Preparing, restructuring, and augmenting a french treebank : Lexicalised parsers or coherent treebanks ? In *Proceedings of PACLING 07*, p. 200–209.

STENETORP P., PYYSALO S., TOPIĆ G., OHTA T., ANANIADOU S. & TSUJII J. (2012). brat : a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 102–107, Avignon, France : Association for Computational Linguistics.

WOODS W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, **13**(10), 591–602.

YIMAM S. M., GUREVYCH I., DE CASTILHO R. E. & BIEMANN C. (2013). Webanno : A flexible,web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, p. 1–6, Stroudsburg, PA, USA : Association for Computational Linguistics.