HAL
open science

Un étiqueteur de rôles grammaticaux libre pour le français intégré à Apache UIMA

Charles Dejean, Manoel Fortun, Clotilde Massot, Vincent Pottier, Fabien Poulard, Matthieu Vernier

To cite this version:

Charles Dejean, Manoel Fortun, Clotilde Massot, Vincent Pottier, Fabien Poulard, et al.. Un étiqueteur de rôles grammaticaux libre pour le français intégré à Apache UIMA. 17e Conférence sur le Traitement Automatique des Langues Naturelles, Jul 2010, Montréal, Canada. hal-00493847

HAL Id: hal-00493847

https://hal.science/hal-00493847v1

Submitted on 21 Jun 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Un étiqueteur de rôles grammaticaux libre pour le français intégré à Apache UIMA

Charles Dejean Manoel Fortun Clotilde Massot Vincent Pottier
Fabien Poulard[1] Matthieu Vernier[1]
(1) LINA, UMR6241, 44322 Nantes
{Fabien.Poulard, Matthieu.Vernier}@univ-nantes.fr

Résumé. L'étiquetage des rôles grammaticaux est une tâche de pré-traitement récurrente. Pour le français, deux outils sont majoritairement utilisés : TreeTagger et Brill. Nous proposons une démarche, ne nécessitant aucune ressource, pour la création d'un modèle de Markov caché (HMM) pour palier les problèmes de ces outils, et de licences notamment. Nous distribuons librement toutes les ressources liées à ce travail.

Abstract. Part-of-speech tagging is a common preprocessing task. For the French language, Brill and TreeTagger are the most often used tools. We propose a method, requiring no resource, to create a Hidden Markov Model to get rid of the problems and licences of these tools. We freely distribute all the resources related to this work.

Mots-clés : étiquetage grammatical, Modèle de Markov caché, UIMA, Brill, TreeTagger.

Keywords: grammatical tagging, Hidden Markov Model, UIMA, Brill, TreeTagger.

1 Introduction et besoins

Dans les travaux en traitement automatique des langues, l'étiquetage des rôles grammaticaux dans les textes est une tâche de pré-traitement récurrente. Les résultats de l'étiquetage servent de support à des tâches plus complexes ou de plus haut niveau linguistique : l'extraction terminologique, la recherche d'informations, la recherche de patrons grammaticaux et sémantiques, la fouille d'opinions, la catégorisation de textes, la détection de dérivation de textes, etc. Autrement dit, de l'efficacité et de la qualité de cette étiquetage dépendent une majorité des problèmes applicatifs ou de recherche. D'une part, les erreurs de résolution des rôles grammaticaux sont répercutées sur les tâches suivantes. D'autre part, la performance des outils d'étiquetage grammatical ont un impact non-négligeable sur le temps de calcul global des traitements. Des temps de calcul trop importants sont un frein au passage à l'échelle des prototypes développés dans un cadre scientifique vers des outils industriels.

Pour le traitement du français, deux outils sont majoritairement utilisés : TreeTagger (Schmid, 1994) et Brill (Brill, 1992). Les motivations qui expliquent notre intérêt pour développer un nouvel étiqueteur de rôles grammaticaux pour le français sont essentiellement de trois types :

– La licence de TreeTagger est restreinte au cadre de la recherche académique. La version UNIX de Brill est distribuée librement mais à notre connaissance les ressources pour le français ne sont pas distribuées

sous une licence permissive, le droit par défaut (restrictif) s'applique alors. Dans ce contexte, il est donc difficile de promouvoir des prototypes d'applications scientifiques dans un contexte industriel sans trouver une alternative et revoir alors son intégration avec le reste de l'application.
– L'utilisation automatisée de Brill et TreeTagger dans une chaîne de composants est d'ailleurs un aspect critique du point de vue de l'interopérabilité. Les formats d'entrée/sortie de ces outils leurs sont propres et requièrent le développement de composants d'encapsulation. Par ailleurs, quelques cas de bogues (ou des temps de calcul très long) existent notamment pour les textes très volumineux, peu ponctués ou encodés de manière non-homogène dans le format attendu par l'outil.
– Enfin, d'un point de vue scientifique, nous souhaitons observer la faisabilité de tirer profit des décisions différentes prises par Brill et TreeTagger pour améliorer la qualité des résultats en confrontant et en apprenant de leurs erreurs. Nous n'utilisons aucune nouvelle ressource pour cela, mais annotons automatiquement un corpus à l'aide de ces derniers.

Dans cet article, nous présentons deux expérimentations. Premièrement, la comparaison de divergences des étiqueteurs Brill et TreeTagger afin de les évaluer comparativement (*cf. section 3.1*). Deuxièmement, l'apprentissage d'un modèle de Markov caché pour le français permettant d'annoter les rôles grammaticaux via l'outil HmmTagger[1], écrit en langage Java et disponible sous la forme d'un composant Apache UIMA[2] (*cf section 3.2*).

2 Brill et TreeTagger

Brill (Brill, 1992) est un des systèmes d'annotation de rôles grammaticaux les plus souvent cité dans les travaux francophones. Brill a fait l'objet d'un entraînement spécifique pour le français (Lecomte & Paroubek, 1998) à partir d'un corpus de 417 370 mots développé à cette occasion, et qui a les caractéristiques suivantes : (i) il n'est pas distribuable, il contient des morceaux de texte dont les ayant droits ne permettent pas la diffusion. (ii) Il est essentiellement composé de textes littéraires en français (*Balzac*, *Zola*, *Dumas*, *Flaubert,* etc.). (iii) Chacun des mots de ces textes a été étiqueté manuellement selon le jeux d'étiquettes choisi pour le français. Brill utilise un lexique et des fichiers de règles. Le lexique contient une liste de mots, chacun d'eux est associé à une liste d'étiquettes classées dans l'ordre décroissant de fréquences d'apparitions dans le corpus annoté. Les règles permettent de déterminer une étiquette probable à partir du contexte quand un mot n'est pas dans le lexique (ex : *DTN :sg PRV :sg NEXTTAG VCJ :sg*).

Le TreeTagger (Schmid, 1994) remplit la même tâche avec un jeu d'étiquettes légèrement différent. Il est constitué de deux parties : l'une pour l'apprentissage d'une langue et l'autre pour l'annotation à proprement parlé. Le système d'apprentissage requiert, pour une langue donnée, un lexique de formes fléchies, un jeu d'étiquettes et un corpus d'entraînement. L'apprentissage effectué par TreeTagger consiste ensuite à évaluer la probablité d'une transition entre un mot (ou sa catégorie grammaticale) et un autre mot (ou sa catégorie grammaticale), puis à générer un arbre de décision binaire à partir des probalités calculées. Pour le français, le jeu est composé de 33 étiquettes grammaticales[3]. La phase d'entraînement permet de produire un lexique contenant la liste des possibilités d'étiquetage pour chaque mot. Il se scinde en trois parties : un lexique de formes fléchies, un lexique de suffixes, une entrée par défaut. Pour chaque mot, le TreeTagger sélectionne d'abord l'étiquette la plus probable selon le lexique des formes fléchies (ou la déduit à partir du suffixe), puis il la corrige en mesurant la probabilité que l'étiquette *c* suive une séquence de deux autres étiquettes *ab*.

[1] `http://uima.apache.org/downloads/sandbox/hmmTaggerUsersGuide/hmmTaggerUsersGuide.html`
[2] `http://uima.apache.org`
[3] `http://www.ims.uni-stuttgart.de/~schmid/french-tagset.html`

Commun	Treetagger	Brill	Définition
ABR	ABR	ABR	Abréviation
ADJ	ADJ		Adjectif
ADJ		ADJ :sg	Adjectif (sauf Participe passé) au singulier
ADJ		ADJ :pl	Adjectif (sauf Participe Passé) au pluriel
NUM	NUM		Numéral
NUM		CAR	Cardinal (en chiffres ou en lettres)

TAB. 1 – Extrait des correspondances considérées entre les jeux d'étiquettes de Brill et TreeTagger

3 Expérimentations

La plateforme UIMA (Unstructured Information Management Architecture) constitue notre *paillasse*. UIMA est un environnement de développement dédié à la structuration de documents initié par IBM et repris par la communauté Apache. Elle permet notamment, grâce à la norme UIMA approuvée par l'OASIS, d'assurer l'échange des données entre les composants d'une même chaîne de traitement et leur interopérabilité, de ré-utiliser des composants existants et de construire des applications robustes complexes destinées à la recherche ou à l'industrie.

Dans la boite à outils de UIMA, le composantHMMTagger[4] implémente un modèle de Markov caché (HMM) qui permet d'annoter les rôles grammaticaux d'un texte préalablement découpé en mots. Le HMMTagger nécessite pour cela un modèle de Markov caché pour la langue à traiter. Il n'existe pas de tel modèle pour le français. Notre objectif est d'entraîner un modèle HMM pour le français et de l'utiliser avec le composant HMMTagger de UIMA. Nous ne disposons d'aucune ressource (corpus annoté, lexique, règles) en amont de ce travail. Dès lors, la méthode envisagée consiste à construire un corpus annoté sans le coût d'une annotation manuelle en (i) annotant un corpus automatiquement à l'aide du TreeTagger et de Brill (encapsulés dans UIMA par le LINA (Hernandez *et al.*, 2010)) ; (ii) en regroupant les jeux d'étiquettes de ces deux outils sous un jeu commun afin de les comparer. Nous profitons de cet objectif principal pour comparer la qualité des annotations de TreeTagger par rapport à Brill, et comparer leurs performances d'un point de vue du temps d'exécution.

Données expérimentales : un corpus et un jeu d'étiquettes commun Nous avons extrait des articles dits *de qualité* sur Wikipédia dans des thématiques variées (31 documents). Ceux-ci correspondent selon Wikipédia à un contenu riche et rédigé dans un bon français. Nous ajoutons à ce corpus, des cours extraits de la Wikiversity (10 documents), et quelques news de Wikinews (3 documents). Le corpus constitué est composé de **496 886** mots, soit une taille similaire à celle du corpus utilisé pour l'entraînement de Brill pour le français. Ces textes ont également l'avantage d'être librement utilisables et distribuables sous licence CC-by-sa 3.0[5] pour Wikiversity et Wikipedia et CC-by 2.5[6] pour Wikinews.

Afin de permettre une évaluation comparative de Brill et TreeTagger, nous alignons au préalable leur jeu

[4] http://uima.apache.org/downloads/sandbox/hmmTaggerUsersGuide/hmmTaggerUsersGuide.html
[5] http://creativecommons.org/licenses/by-sa/3.0/deed.fr
[6] http://creativecommons.org/licenses/by/2.5/deed.fr

	Exp. A	Exp. B
Nombre prédictions évaluées	4453	2805
	(0,89% corpus ; 3,17% incohérences)	(0,55% du total ; 3,57% incohérences)
Nombre total d'évaluations	5486	4921
... dont erreur de Brill	961 (17,52 %)	739 (15,02 %)
... dont erreur de TTg(A)/HMM(B)	3145 (57,33 %)	3177 (64,56 %)
... dont erreur des deux	801 (14,60 %)	618 (12,56 %)
... dont erreur dû à la tokenisation	579 (10,55 %)	387 (7,86 %)
Évaluations qui se recouvrent	887	460
Kappa	0,99	0,96

TAB. 2 – Synthèse des évaluations comparatives entre Brill et TreeTagger (Exp. A) et Brill et HMMTagger (Exp. B) sur le jeu d'étiquettes commun.

d'étiquettes respectif pour établir un socle commun. Le tableau 1 présente un extrait[7] de cette table de correspondances. Il n'y a pas de problème d'ambiguïté, il s'agit en général de renommer des étiquettes ou d'enlever un niveau de spécificité propre à TreeTagger ou Brill selon les cas (par exemple, contrairement à TreeTagger, Brill ajoute l'information du *nombre* sur un adjectif).

3.1 Expérience A : Comparaison de TreeTagger et Brill

Évaluation des incohérences entre Brill et TreeTagger Une fois la projection vers le jeu commun réalisée, deux cas peuvent se présenter pour un mot donné. Premièrement, les prédictions de Brill et de TreeTagger sont cohérentes : les deux étiquettes sont identiques. Ceci ne garantit pas la justesse de la prédiction, mais montre que, pour le mot concerné, les deux outils concordent et qu'ils sont corrects ou bien en erreur tous les deux. Deuxièmement, les prédictions des deux outils sont incohérentes : les étiquettes sont différentes. Dans cette configuration, il est nécessaire d'évaluer manuellement. Sur les 496 886 mots extraits de notre corpus, les outils sont en désaccord sur 103 598 (env. 20,30 %) soit un peu plus de 5 mots par phrase. Six annotateurs humains ont évalué ces incohérences, en connaissance du mot dans son contexte et des étiquettes du jeu commun posées par les deux outils. Ils pouvaient décider si l'un ou l'autre était correct, si les deux avaient tort ou bien si l'erreur avait été provoquée par le découpage en mots. Au total l'évaluation manuelle des 4 453 mots, soit 3,17 % des incohérences identifiées, a été répartie entre les annotateurs. L'accord inter-annotateur est excellent : l'indice Kappa est de 0,99. Il a été calculé sur 887 instances évaluées communément par deux des six annotateurs (*cf. Tableau 2*).

Comparaison des résultats Habituellement, l'évaluation d'un outil se fait en absolu : on évalue l'outil sur l'ensemble de son fonctionnement afin d'obtenir un score estimant le nombre d'erreurs ou de succès. Dans le cas présent, nous voulons évaluer les outils l'un par rapport à l'autre, nous avons donc besoin d'une métrique permettant de comparer les performances des deux outils sans connaître leur nombre d'erreurs sur l'intégralité du corpus. Tout d'abord nous supposons que les résultats du Tableau 2 sont valables pour toutes les incohérences (140 400 sur les 496 886 mots du corpus). Ainsi, le taux d'erreur de Brill sur les incohérences serait de $\delta_{Brill} = 17,52\% + 14,60\% + 10,55\% = 42,67\%$ (erreurs de Brill, erreurs des deux

[7] Le tableau de correspondances complet est disponible dans les ressources distribuées

et erreurs dû à la tokenisation). Sur le reste du corpus, les prédictions étant cohérentes, le taux d'erreur de Brill et de TreeTagger est identique et égal à α. Le taux d'erreur de Brill sur le corpus complet est donc, avec C_c et C_i respectivement le nombre d'éléments cohérents et incohérents, $\Delta_{Brill} = \frac{(\alpha * C_c)+(\delta_{Brill} * C_i)}{C_c+C_i}$, de manière similaire $\Delta_{TTg} = \frac{(\alpha * C_c)+(\delta_{TTg} * C_i)}{C_c+C_i}$, soit encore $\frac{\Delta_{TTg}}{\Delta_{Brill}} = \frac{(\alpha * C_c)+(\delta_{TTg} * C_i)}{(\alpha * C_c)+(\delta_{Brill} * C_i)}$. Or $0 \leq \alpha \leq 100$, donc $\Delta_{TTg} = \beta \Delta_{Brill}$ avec $1.13 \leq \beta \leq 1.93$. En d'autres termes, le taux d'erreur de TreeTagger est entre 1,13 fois et 1,93 fois plus élevé que celui de Brill. Ce résultat est notamment pertinent pour l'évaluation des performances du HMMTagger (*cf. section 3.2*).

Fusion plus fine des étiquettes vers le jeu commun À partir des évaluations manuelles des incohérences, nous avons repris la méthode d'annotation automatique du corpus : lorsque Brill et TreeTagger s'accordent sur une étiquette nous la retenons, lorsqu'ils sont en désaccord nous appliquons des règles ad-hoc issues de l'évaluation manuelle. Cette fusion plus fine nous a permis de réduire le nombre d'incohérences de près de 25 %, passant ainsi de 103 598 à 78 578.

Les règles de résolution sont issues de règles d'associations[8] extraites de l'ensemble des évaluations manuelles. Chaque mot dont l'étiquette a pu être corrigée lors de l'évaluation a été caractérisé par les étiquettes du jeu commun posées par Brill et TreeTagger, leurs étiquettes propres et le mot lui-même (p. ex. *TagOrigBrill=ECJ :sg TagOrigTTG=VER :pres* → *TagRetenu=VER :ET*). Nous n'avons conservé que les règles dont la confiance est de 1.0 afin de limiter l'introduction d'erreurs. Sur la soixantaine de règles extraites, nous en avons implémenté 11, les autres règles subsumant les premières.

3.2 Expérience B : Évaluation comparative du modèle HMM français avec Brill

À des fins d'expérimentations, nous scindons notre corpus en deux (90 %/10 %) : un corpus d'apprentissage, et un corpus de test pour l'évaluation comparative de Brill et HMMTagger.

Apprentissage du modèle HMM pour le français Il est à noter que 15 % des mots ne sont pas étiquetés lorsque Brill et TreeTagger ne sont pas d'accord et qu'aucune règle de fusion pertinente n'a pu être mise en oeuvre. Nous avons choisi de ne pas adopter de stratégie par défaut de type *dans le doute, faire confiance à Brill*, pour ne pas biaiser notre approche. Le module d'apprentissage du HMMTagger utilise ensuite les mots étiquetés après fusion (mot + étiquette) pour générer un modèle HMM.

Tests et évaluation À l'image de l'expérimentation A (*cf. section 3.1*), nous effectuons une évaluation comparative entre le HMMTagger développé et Brill, que nous considérons alors comme une meilleure référence que TreeTagger. Nous appliquons sur les résultats présentés dans le Tableau 2 la métrique discutée précédemment. Ainsi, le taux d'erreur du HMMTagger ($\frac{\Delta_{HMM}}{\Delta_{Brill}}$) est entre 1,03 fois et 2,4 fois plus élevé que celui de Brill.

4 Discussion

Les résultats de l'évaluation comparative entre Brill et TreeTagger puis entre Brill et le HMMTagger sont dans la même fourchette (respectivement $[1,13..1,93]$ et $[1,03..2,4]$). Brill est donc meilleur que les deux

[8] Les règles d'association ont été calculées dans Weka à l'aide de l'algorithme *weka.associations.Apriori*

Outil	Treetagger	Brill	HMMTagger
Temps de traitement sur notre corpus	5 min 42 s.	6 min 51 s.	11 s.

TAB. 3 – Performance des composants Brill, TreeTagger et HMMTagger sur l'annotation du corpus test.

autres outils, le HMMTagger faisant potentiellement un peu plus d'erreurs que le TreeTagger. Le choix de ne pas apprendre les annotations incohérentes a privé le HMMTagger de la connaissance de mots fréquents pour lesquels il se résigne à un choix par défaut (p. ex. la préposition *à*, les flexions du verbe *avoir*, ...). L'implémentation de règles manuelles pour ces quelques cas devrait permettre d'améliorer significativement les résultats.

Les outils TreeTagger et Brill utilisés dans ces expérimentations sont des encapsulations en composants UIMA des programmes originaux (Hernandez *et al.*, 2010). De telles encapsulations posent des problèmes d'interopérabilité (les programmes de Brill et TreeTagger ne sont pas présents sur toutes les plateformes), et de performances (chargement des ressources et des programmes en mémoire à chaque traitement). Au contraire le HMMTagger, nativement implémenté en Java et intégré à UIMA, est de 29 à 35 fois plus rapide que les outils encapsulés (*cf. Tableau 3*).

D'après Karl Popper (Popper, 1963), « une théorie n'est scientifique si et seulement si elle peut être réfutée ». Cette réfutabilité n'est possible qu'en réitérant les expérimentations. Dans ce sens, nous attachons une importance particulière à distribuer librement nos ressources (le modèle, le code source et le corpus permettant de le générer, ainsi que les résultats intermédiaires de nos expérimentations)[9]Nous encourageons d'ailleurs la généralisation de cette démarche dans le cadre des travaux scientifiques en traitement automatique des langues.

Références

BRILL E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, p. 152–155, Morristown, NJ, USA : Association for Computational Linguistics.

HERNANDEZ N., POULARD F., VERNIER M. & ROCHETEAU J. (2010). New challenges for nlp frameworks. In *Proceedings of New Challenges for NLP Frameworks workshop in LREC'10*, Marrakech, Morocco : European Language Resources Association (ELRA).

LECOMTE J. & PAROUBEK P. (1998). Le catégoriseur d'Eric BRILL. mise en oeuvre de la version entraînée à l'INALF, rapport technique. Nancy.

POPPER K. (1963). *Conjectures and Refutations*.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, p. 44–49.

[9]Tous ces éléments sont archivés à l'adresse `http://recherche.fabienpoulard.info/taln2010`. Le composant HMMTagger pour UIMA est disponible à l'adresse `http://uima.apache.org/sandbox.html#tagger.annotator`, il suffit de le configurer pour utiliser notre modèle *hmmtagger_model_fr_20100501.dat* présent dans l'archive.