Table des matières

Articles longs

Fouille de données et TAL

Sémantique

Parsing 1

Lexique 1

Gestion des erreurs en TAL

Modèles linguistiques

Méthodes numériques pour le TAL

Lexique 2

Étiquetage 1

Traitement de corpus 1

Sentiments

Outils

Étiquetage 2

Langue des signes

Résumé automatique

Traduction automatisée d'une oeuvre littéraire: une étude pilote

Laurent Besacier[1]
(1) LIG, Université de Grenoble, UJF - BP 53, 38041 Grenoble Cedex 9
laurent.besacier@imag.fr

Résumé. Les techniques actuelles de traduction automatique (TA) permettent de produire des traductions dont la qualité ne cesse de croitre. Dans des domaines spécifiques, la post-édition (PE) de traductions automatiques permet, par ailleurs, d'obtenir des traductions de qualité relativement rapidement. Mais un tel pipeline (TA+PE) est il envisageable pour traduire une oeuvre littéraire ? Cet article propose une ébauche de réponse à cette question. Un essai de l'auteur américain Richard Powers, encore non disponible en français, est traduit automatiquement puis post-édité et révisé par des traducteurs non-professionnels. La plateforme de post-édition du LIG utilisée permet de lire et éditer l'oeuvre traduite en français continuellement, suggérant (pour le futur) une communauté de lecteurs-réviseurs qui améliorent en continu les traductions de leur auteur favori. En plus de la présentation des résultats d'évaluation expérimentale du pipeline TA+PE (système de TA utilisé, scores automatiques), nous discutons également la qualité de la traduction produite du point de vue d'un panel de lecteurs (ayant lu la traduction en français, puis répondu à une enquête). Enfin, quelques remarques du traducteur français de R. Powers, sollicité à cette occasion, sont présentées à la fin de cet article.

Abstract. Current machine translation (MT) techniques are continuously improving. In specific areas, post-editing (PE) allows to obtain high-quality translations relatively quickly. But is such a pipeline (MT+PE) usable to translate a literary work (fiction, short story) ? This paper tries to bring a preliminary answer to this question. A short story by American writer Richard Powers, still not available in French, is automatically translated and post-edited and then revised by non-professional translators. The LIG post-editing platform allows to read and edit the short story suggesting (for the future) a community of readers-editors that continuously improve the translations of their favorite author. In addition to presenting experimental evaluation results of the pipeline MT+PE (MT system used, auomatic evaluation), we also discuss the quality of the translation output from the perspective of a panel of readers (who read the translated short story in French, and answered to a survey afterwards). Finally, some remarks of the official french translator of R. Powers, requested on this occasion, are given at the end of this article.

Mots-clés : traduction automatique, TA, oeuvre littéraire, post-édition.

Keywords: machine translation, MT, litterature, fiction, post-edition.

1 Introduction

La tâche de post-édition consiste à éditer la sortie textuelle produite, le plus souvent par une machine (système de traduction automatique, de reconnaissance optique de caractères, de transcription automatique, etc.) en vue de l'améliorer. Dans le domaine de la traduction de documents, où les systèmes automatiques sont utilisés, le processus suivant est généralement utilisé : le système de TA produit des traductions brutes qui sont manuellement post-édités par des traducteurs professionnels formés (post-éditeurs) corrigeant les erreurs de traduction.

De nombreuses études ont montré les avantages de l'utilisation combinée traduction automatique+post-edition manuelle (TA+PE) dans un flux de diffusion. Par exemple, Garcia (2011) a montré que même si la post-édition de sorties brutes de TA ne conduit pas toujours à une amélioration en termes de productivité, elle permet de produire de meilleures traductions par rapport à des traductions manuelles directement issues du texte source [1]. Autodesk a aussi réalisé une expérience pour tester si l'utilisation de la TA permettrait d'améliorer la productivité des traducteurs. Dans cette expérience, les résultats [2] montrent que la post-édition des sorties issues d'un système automatique augmente significativement la productivité par rapport à une traduction partant "de zéro". Ceci est vérifié quelle que soit la paire de langues, l'expérience du traducteur et sa préférence exprimée concernant la méthode (post-éditer des sorties automatiques *vs* traduire à partir de zéro).

1. le travail de Garcia (2011) est cependant sujet à controverses car la traduction manuelle sans post-édition semble avoir été faite sans autoriser l'utilisation d'aide numérique pour assister le traducteur (même pas un dictionnaire électronique)
2. amta2012.amtaweb.org/AMTA2012Files/html/5/5_paper.pdf

Ces résultats mesurés en milieu académique (Garcia (2011)) ou industriel (Autodesk) sur des processus de traduction dans des domaines spécialisés, nous conduisent à poser la question suivante : quelle serait la valeur d'un tel processus (TA+PE) appliqué sur la traduction d'une oeuvre littéraire ? Combien de temps faut-il pour traduire une oeuvre de dix mille mots ? La traduction obtenue est-elle acceptable pour des lecteurs ? Qu'en penserait le traducteur attitré de l'auteur considéré ? Une traduction "low cost" produite par des communautés de fans (comme c'est le cas pour les séries TV) est-elle envisageable pour des romans ou des nouvelles ? Cet article tente d'apporter de premières réponses (certes très partielles) à ces questions. En plus de cette étude préliminaire, il restera au moins de ce travail la traduction inédite d'une nouvelle, ou plutôt d'un essai (*The Book of Me* de Richard Powers).

Le suite de l'article présente notre méthodologie générale, le choix de l'oeuvre et les systèmes de TA et de PE (tous deux issus du LIG) utilisés. Les données expérimentales collectées (et rendues disponibles à la communauté sur *github*) sont ensuite décrites dans la section 3. Dans la partie 4, nous tentons d'aller au delà d'un tâche de TA adaptée au domaine en sollicitant le point de vue d'un (petit) panel de lecteurs et l'avis du traducteur attitré de R. Powers. Pour finir, la partie 5 conclut ce travail préliminaire.

2 Traduction automatisée d'une oeuvre littéraire

2.1 Remarques préliminaires

Le format court de cet article ne permet pas de proposer une étude bibliographique exhaustive sur la traduction assitée de documents littéraires ou, plus généralement, sur l'utilisation du TALN dans le domaine littéraire. Cependant, on peut noter que c'est un champ de recherche assez actif : un atelier y est consacré depuis 2012 : *ACL workshop on Computational Linguistics for Literature* [3]. On peut aussi mentionner la tenue récente, en France, de la *Journee d'étude sur la traduction de textes litteraires* organisée par l'ENS en 2013. Cependant, à notre connaissance, cette étude est la première sur la traduction automatisée d'une oeuvre littéraire. Il conviendrait également de définir ce qu'on appelle un *texte littéraire*. En ce qui nous concerne, nous incluons dans cette catégorie (notre définition est sans doute trop restrictive) toutes les oeuvres de fiction ou autobiographiques écrites sous la forme de romans ou de nouvelles. Dans ces textes, l'auteur exprime sa vision du monde, de son époque et de la vie en général tout en utilisant des *procédés littéraires* et une *technique d'écriture* (forme) lui permettant de créer des effets à l'aide du langage ou de faire passer des messages (explicites ou sous-entendus).

2.2 Choix de l'oeuvre

Le choix de l'oeuvre a été guidé par le fait que (a) nous avions un contact avec le traducteur français de l'auteur américain Richard Powers [4] (auteur notamment du roman *La chambre aux échos* qui a remporté le National Book Award et a été finaliste du Prix Pulitzer) (b) R. Powers publie des romans souvent en lien avec la science et, d'une certaine manière, ses textes contiennent quelques points communs avec des textes scientifiques ou techniques (ce qui réduit peut-être un peu le fossé entre traduction de textes scientifiques et littéraires ?).

Par l'intermédiaire de son traducteur français (J-Y Pellegrin), R. Powers a été informé, par e-mail, de notre démarche et il a donné son accord. Nous avons ensuite choisi une oeuvre, non encore traduite en français, intitulée *The Book of Me* et publiée pour la première fois dans les colonnes du magazine GQ [5]. C'est un texte narratif, raconté à la première personne, où l'auteur est le personnage principal de l'histoire. Il y raconte l'année 2008 au cours de laquelle il est devenu la neuvième personne au monde à voir son génome entièrement séquencé. Bien que le thème soit la génétique et malgré le style simple et clinique employé par l'auteur, *The Book of Me* est bien une oeuvre littéraire où l'auteur, qui enseigne la technique narrative à l'université, ne met jamais de côté son ambition poétique, son humour et sa fascination pour l'irrationnel.

2.3 Méthodologie générale

La traduction de textes par cascade TA+PE (traduction automatique puis post-édition) a déjà été évaluée au LIG sur des données journalistiques. En 2012, la collecte de post-éditions manuelles de 12 000 énoncés (équivalente à un livre de 500 pages environ) a été réalisée par Potet *et al.* (2012). Ce corpus collecté via *crowdsourcing* et disponible en ligne [6], est un des plus gros corpus existant concernant la post-édition de données issues d'un système de traduction automatique appliqué sur des données libres de droit. Il est par exemple trois fois plus important que celui collecté par Specia *et al.* (2010) qui fait référence dans le domaine. La même méthodologie a été appliquée sur l'oeuvre littéraire choisie mais

3. sites.google.com/site/clfl2014a/
4. fr.wikipedia.org/wiki/Richard_Powers
5. www.gq.com/news-politics/big-issues/200810/richard-powers-genome-sequence
6. www-clips.imag.fr/geod/User/marion.potet/index.php ?page=download

(différence importante) il n'y avait qu'un seul post-éditeur (pas de *crowdsourcing*) et le texte postédité a été ensuite révisé.

Plus précisément, le corpus a été divisé en trois parties égales. Une boucle *traduction-postédition-adaptation* est appliquée sur les trois blocs de texte selon le processus suivant :
– Après traduction automatique du premier tiers (par un système de TA probabiliste initial non adapté à la tâche), les sorties traduites sont révisées par une étudiante en Master de traduction,
– Ce premier tiers post-édité sert à adapter le système de TA Anglais-Français. Vu la faible taille des données post-editées, seuls les poids du modèle log-linéaire sont adaptés en utilisant le premier tiers corrigé comme corpus de développement. Une méthode similaire est suggérée par Pecina *et al.* (2012) pour l'adaptation au domaine avec quantité de données limitées (nous sommes conscients que d'autres techniques d'adaptation plus avancées auraient pu être utilisées mais ceci n'était pas le thème central de notre contribution),
– Ensuite, le second tiers de l'oeuvre est traduit avec le système de TA adapté, puis la sortie est post-éditée et un second système de TA adapté est obtenu à partir de ces nouvelles données. Ce second système est utilisé pour traduire la troisième et dernière partie de l'oeuvre,
– Une fois la PE terminée, le texte final est révisé : d'abord par le post-éditeur puis par un autre réviseur (non professionel, francophone, ayant cependant une bonne connaissance de la langue anglaise),
– Les temps de post-édition et de révision sont mesurés.

2.4 Système de TA utilisé

Le système LIG est entraîné à partir des données fournies lors de la campagne d'évaluation IWSLT et optimisé pour la tâche de TA anglais-français de la campagne 2012. Les corpus suivants sont utilisés pour construire le modèle de traduction (total cumulé de 25M phrases environ) : *news-c, europarl, un, ted, eu-const, dgt-tm, pct* et une extraction de 5M de phrases du corpus *gigaword*. La partie en français des mêmes corpus est utilisée pour apprendre le modèle de langue, avec l'ajout du corpus *news-shuffle* fourni au cours de la campagne WMT 2012. Le système LIG est un système à base de segments (*phrase-based*) qui s'appuie sur la boîte à outils *Moses* Hieu *et al.* (2007). Trois modèles de traduction appris à partir de différents corpus (*ted ; news-c+europarl+un+eu-const+dgt-tm+pct ; gigaword5M*) sont utilisés. Un modèle de langue 5-gramme est appris séparément sur chaque corpus à l'aide de la boîte à outils SRILM Stolcke (2002) avec un modèle de repli Kneser-Ney modifié, puis les modèles sont interpolés en optimisant la perplexité sur le corpus *dev2010* de la campagne IWSLT. Le système du LIG obtient des scores BLEU de 36,88 et 37,58 sur les corpus tst2011 et test2012 de IWSLT respectivement (BLEU évalué avec casse et ponctuation). Plus de détails sur ce système (classé honorablement lors de IWSLT 2012) se trouvent dans Besacier *et al.* (2012). Il est clair que ce système n'est pas optimal pour traduire des textes littéraires et il serait souhaitable, pour de futurs travaux, de rassembler au moins des textes littéraires en français pour adapter le modèle de langue cible (ou même envisager d'avoir accès aux autres œuvres et traductions de l'auteur).

2.5 Post-edition

Nous utilisons l'interface de post-édition du LIG proposée initialement par Huynh *et al.* (2008). Celle-ci a été utilisée pour de nombreux projets : traduction d'articles de l'encyclopédie EOLLS, accès multilingue à des dizaines de sites Web. La figure 1 montre l'interface de post-édition en mode avancé qui permet, pour chaque segment source, de charger plusieurs traductions automatiques (issues de Google, Moses [7] , etc.) et de les corriger. Le temps de post-édition est mesuré et l'historique des corrections est stocké dans une base de données. Cet outil est la colonne vertébrale ayant donné lieu au concept d'iMAG proposé par le LIG (*interactive Multilingual Access Gateway*) qui permet la navigation dans un site ou sous-site Web dans plusieurs langues d'accès, avec amélioration incrémentale et contrôlée de la qualité des traductions (voir le lien en note de bas de page [8] pour plus de détails).

3 Données expérimentales

3.1 Corpus et statistiques de post-édition

L'oeuvre, composée de 545 segments et 10731 mots est divisée en trois blocs identiques. La Table 1 résume le nombre de mots des données source et cible (TA ou PE [9]). Sans surprises, un ratio supérieur à 1,2 est observé entre cible française (TA) et source anglaise. On constate cependant que ce ratio tend à diminuer après post-édition de la sortie française.

7. ici, la sortie de notre système était chargée en priorité par rapport à Google
8. aximag.fr/AXiMAG-homePage.html
9. La post-édition utilisée ici est obtenue après chaque itération du processus ; la dernière étape de révision n'est donc pas prise en compte à ce stade.

FIGURE 1 – Interface de post-édition en mode avancé

TABLE 1 – Corpus source, cible traduite et cible corrigée

Itération (nb. seg)	Anglais (nb. mots)	TA Français (nb. mots)	PE Français (nb. mots)
It.1 (184)	3593	4295	4013
It.2 (185)	3729	4593	4202
It.3 (176)	3409	4429	3912
Total (545)	**10731**	**13317**	**12127**

3.2 Performances du système de TA

La Table 2 résume les performances de TA (mesurées avec BLEU) calculées sur le corpus complet avec les systèmes issus de chaque itération. Le temps de post-édition requis pour chaque bloc est également indiqué. Les scores BLEU, qui sont directement comparables, ne montrent pas de véritable amélioration du système. Il semble donc que l'adaptation des poids seuls (qui donne lieu à des améliorations dans Pecina *et al.* (2012)) soit peu efficace dans notre cas. Le temps de post-édition diminue cependant légèrement à chaque itération (mais là encore, les différences sont faibles et il est difficile de dire si la diminution du temps de PE est dûe à l'adaptation du système de TA ou à une productivité croissante du post-editeur qui s'adapte à la tâche). Au final, le temps total de PE est estimé à 15h environ.

TABLE 2 – Evaluation automatique (BLEU) sur corpus complet et mesure du temps de PE pour chaque bloc de l'itération

Système TA utilisé	score BLEU (corpus complet)	temps PE (bloc it.)
It.1 (non adapté)	38,95	5h37mn
It.2 (tuning sur bloc 1)	37,51	4h45mn
It.3 (tuning sur bloc 1+2)	39,38	4h35mn

La lecture de l'oeuvre traduite à ce stade (après PE) est peu satisfaisante. En effet, la post-édition est faite "segment par segment", sans vue d'ensemble sur le texte. Il en résulte une manque d'homogénéité très gênant pour un texte littéraire. Pour cette raison, deux révisions du texte traduit sont également effectuées : une par le post-éditeur lui même (4h) et une par l'auteur de cet article (6h). La version finale de l'oeuvre traduite (qui aura donc été obtenue en 15+4+6=25h) sert de base aux évaluations, plus qualitatives, présentées dans la partie suivante. Les différences entre la version post-éditée brute (15h de travail) et la version révisée (25h de travail) ne sont pas analysées ici, mais pourront l'être dans le futur puisque les deux versions sont rendues disponibles sous *github*.

4 Aller au delà d'une tâche de TA adaptée au domaine

4.1 Le point de vue des lecteurs sur la traduction post-éditée

Neuf lecteurs ont accepté de lire l'oeuvre traduite et ont répondu à un questionnaire, toujours ouvert sur *fluidsurveys.com* [10]. La version *pdf* de l'essai traduit ainsi que fichier tableur rassemblant les résultats du sondage sont également rendus disponibles dans *github*. Après trois questions permettant de mieux cerner le profil du lecteur, une première partie (5 questions) interroge les lecteurs sur la lisibilité et la qualité du texte littéraire traduit. Une seconde partie (7 questions) vérifie que certaines subtilités du texte ont été bien comprises.

Le texte est jugé globalement lisible (5 TBien et 3 Bien), compréhensible (8 oui, 1 non) et contenant peu de fautes (8 rarement, 1 souvent). Les questions de compréhension les plus faciles sont bien traitées par les lecteurs qui répondent tous correctement (4 questions). Cependant, trois questions donnent lieu à des réponses différentes selon les lecteurs :

– 2 lecteurs ont mal répondu à une question apparemment simple (*qui finance le séquençage du génôme de Powers ?*),
– la question *Lors de l'écriture de la nouvelle, combien de personnes se sont déjà fait séquencer le génome ?* était ambigue puisqu'on pouvait répondre 8 ou 9 (en comptant ou non Powers) et a donné lieu à des réponses différentes des lecteurs
– seuls 4 lecteurs sur 9 ont su donner une chronologie correcte des étapes du processus de séquençage d'un génôme ; le texte traduit ne contient cependant pas de contresens ; ce résultat mitigé indique peut-être un désintérêt de certains lecteurs pour les aspects les plus techniques de l'oeuvre.

En résumé, nous pouvons dire que ce sondage, qui reste très limité, montre toutefois que le texte (produit selon notre méthodologie) a été jugé acceptable et assez lisible par nos lecteurs (dont 3 indiquent lire très souvent, 4 assez souvent et 2 rarement). Nous citons également quelques remarques mises dans les commentaires libres : *"J'ai remarqué peu de fautes, quelques néologismes (j'ai considéré que c'étaient des néologismes et pas des erreurs de traduction car ça avait du sens)" - "Texte très fluide et lecture très facile malgré des termes scientifiques précis" - "J'ai trouvé le texte un peu difficile parce qu'il contient des mots complexes et qu'il traite d'un domaine que je ne connais pas du tout"*.

4.2 Le point de vue du traducteur de R. Powers

Pour finir cette étude pilote, un dixième lecteur a été sollicité : le traducteur français de l'auteur, J-Y Pellegrin, enseignant chercheur à Paris-Sorbonne. Son avis est résumé ici sous la forme de questions réponses. Le manque de place ne nous permet pas de commenter ces remarques mais nous pensons qu'elles sont assez explicites pour être délivrées en l'état.

Lisibilité ? *"Le texte auquel vous êtes parvenu restitue une image fidèle du contenu de l'article de Powers. Le pari de la lisibilité est gagné et certains passages (notamment ceux qui portent sur les aspects scientifiques de l'expérience décrite) sont très convaincants."*

Imperfections ? *"Il reste bien sûr des imperfections, des lourdeurs, voire des erreurs ponctuelles, qui appellent une correction"*

Principales erreurs ?

– *"Le défaut le plus répétitif, celui dont souffre d'ailleurs le travail de tout traducteur débutant, est le calque syntaxique, là où le français structure différemment la phrase .../... On comprend, mais ça ne sonne pas vraiment français"*
– *"Autre défaut assez fréquent, la perte des idiomatismes du français au profit d'anglicismes. Parfois ces anglicismes peuvent être plus dérangeants lorsqu'ils flirtent avec le franglais comme dans « connaissances actionnables » (p. 18) au lieu de « connaissances pratiques / utilisables ». "*
– *"Un troisième défaut tient à la non prise en compte de certains repères culturels .../... Par exemple, Powers fait plusieurs références à la topographie de Boston qui donnent lieu à des inexactitudes dans la traduction : « la rivière Charles » par exemple (p. 12) qui n'est pas une rivière mais plutôt un fleuve ; c'est pourquoi on traduira par « la Charles River » ou simplement « la Charles »"*

Ce texte pourrait-il servir de base de départ à un traducteur littéraire professionnel ?" *"Instinctivement, je serais tenté de répondre non pour l'instant, parce que, dès son premier jet, le traducteur possède des réflexes qui lui permettent de produire un texte plus « propre » que celui auquel vous êtes parvenu .../... Cependant, ce traducteur passera plus de 25 heures à produire les 42 feuillets de 1500 signes correspondant au texte de Powers. À raison de 7 feuillets par jour en moyenne, il faut 6 journées de 8h pour venir à bout du texte .../... Si, en revanche, je pouvais ne travailler que sur votre texte, en oubliant complètement celui de Powers parce que j'aurais la garantie que votre traduction ne comporte aucune erreur, ni oubli, ni aplatissement par rapport à l'original, mais qu'elle demande simplement à être améliorée, rendue plus fluide, dans un français plus authentique, les choses seraient différentes et le gain de temps sans doute considérable."*.

10. https ://fluidsurveys.com/surveys/manuela-cristina/un-livre-sur-moi-qualite-de-la-traduction/ ?TEST_DATA=

5 Conclusion

5.1 Données rassemblées et mise en ligne

Les données de cet article sont disponibles sur le lien suivant *github.com/powersmachinetranslation/DATA*. On y retrouve notamment :
– les 545 segments source, cible (TA et PE) mentionnés table 1,
– l'oeuvre traduite et révisée en français, ayant été lue par un panel de 9 lecteurs,
– les résultats du questionnaire (9 lecteurs) compilés dans un tableur.

5.2 Commentaires et questions ouvertes

Nous avons présenté une première expérience de traduction automatisée d'une oeuvre littéraire (essai en anglais d'une vingtaine de pages). Les résultats issus d'un pipeline TA+PE ont été présentés et, pour aller au delà, les avis d'un panel de lecteurs et d'un traducteur ont été sollicités. Le texte traduit, obtenu après 25h de travail humain, est jugé acceptable par les lecteurs mais l'avis du traducteur professionnel reste mitigé. Cette approche suggère une méthodologie de traduction rapide et "low cost", analogue aux traductions de sous-titres de séries TV trouvées sur le Web. Pour l'auteur, c'est la possibilité d'avoir son oeuvre traduite dans un plus grand nombre de langues (plusieurs dizaines au lieu d'une poignée - cet essai de R. Powers a d'ailleurs aussi été traduit en roumain avec la même méthodologie). Mais celui-ci est il prêt à sacrifier la qualité de traduction (et son contrôle sur celle-ci) au prix d'une diffusion plus large de ses oeuvres ?

Pour le lecteur qui ne peut lire l'auteur en langue source, c'est la possibilité d'avoir accès plus rapidement à une traduction (certe imparfaite) de son auteur favori. Pour le lecteur non natif mais capable de lire l'oeuvre en langue source, c'est la possibilité d'avoir une aide sur les parties qu'il a du mal à comprendre. Une dernière chose : le titre de l'oeuvre *The Book of Me* est resté inchangé sur la version française car aucune traduction satisfaisante n'a été trouvée pour illustrer le fait que *book* fait référence ici à un livre mais aussi à l'ADN de l'auteur ; ce paradoxe illustre bien toute la difficulté de la traduction d'une oeuvre littéraire.

Remerciements

Merci à Manuela Barcan qui a assuré la première phase de post-édition des traductions automatiques en français et en roumain au cours de l'été 2013. Merci à J-Y Pellegrin, traducteur français de Richard Powers, pour son aide et son ouverture d'esprit. Merci à V. Bellynck et C. Boitet pour leur aide avec l'outil de post-édition Sectra_W.

Références

BESACIER L., LECOUTEUX B., AZOUZI M. & LUONG NGOC Q. (2012). The LIG English to French Machine Translation System for IWSLT 2012. In *In proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*.

GARCIA I. (2011). Translating by post-editing : is it the way forward ? *Journal of Machine Translation*, **25**(3), 217–237.

HIEU H., BIRCH A., CALLISON-BURCH C., ZENS R., AACHEN R., CONSTANTIN A., FEDERICO M., BERTOLDI N., DYER C., COWAN B., SHEN W., MORAN C. & BOJAR O. (2007). Moses : Open source toolkit for statistical machine translation. In *ACL'07, Annual Meeting of the Association for Computational Linguistics*, p. 177–180, Prague, Czech Republic.

HUYNH C.-P., BOITET C. & BLANCHON H. (2008). Sectra_w.1 : an online collaborative system for evaluating, post-editing and presenting mt translation corpora. In *LREC'08, Sixth International Conference on Language Resources and Evaluation*, p. 28–30, Marrakech, Morocco.

PECINA P., TORAL A. & VAN GENABITH J. (2012). Simple and effective parameter tuning for domain adaptation of statistical machine translation. In *Proceedings of the 24th International Conference on Computational Linguistics*, p. 2209–2224, Mumbai, India : Coling 2012 Organizing Committee.

POTET M., EMMANUELLE E R., BESACIER L. & BLANCHON H. (2012). Collection of a large database of french-english smt output corrections. In *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

SPECIA L., CANCEDDA N. & DYMETMAN M. (2010). A dataset for assessing machine translation evaluation metrics. In *7th Conference on International Language Resources and Evaluation (LREC-2010)*, p. 3375–3378, Valetta, Malta.

STOLCKE A. (2002). Srilm : An extensible language modeling toolkit. In *ICSLP'02, 7th International Conference on Spoken Language Processing*, p. 901–904, Denver, USA.

Vers un développement plus efficace des systèmes de traduction statistique : un peu de vert dans un monde de BLEU

Li Gong[1,2] Aurélien Max[1,2] François Yvon[1]
(1) LIMSI-CNRS, Orsay, France (2) Univ. Paris-Sud, Orsay, France
{li.gong,aurelien.max,francois.yvon}@limsi.fr

Résumé. Dans cet article, nous montrons comment l'utilisation conjointe d'une technique d'alignement de phrases parallèles à la demande et d'estimation de modèles de traduction à la volée permet une réduction en temps très notable (jusqu'à 93% dans nos expériences) par rapport à un système à l'état de l'art, tout en offrant un compromis en termes de qualité très intéressant dans certaines configurations. En particulier, l'exploitation immédiate de documents traduits permet de compenser très rapidement l'absence d'un corpus de développement.

Abstract. In this article, we show how using both on-demand alignment of parallel sentences and on-the-fly estimation of translation models can yield massive reduction (up to 93% in our experiments) in development time as compared to a state-of-the-art system, while offering an interesting tradeoff as regards translation quality under some configurations. We show in particular that the absence of a development set can be quickly compensated by immediately using translated documents.

Mots-clés : traduction automatique statistique ; développement efficace ; temps de calcul.

Keywords: statistical machine translation ; efficient development ; computation time.

1 Introduction et motivations

Le développement de systèmes de traduction automatique statistiques repose sur des quantités de données parallèles en constante augmentation (voir par ex. (Smith *et al.*, 2013)). Son cout devient donc très important, en termes de temps de calcul et de moyens de calcul et de stockage. Une conséquence est que la disponibilité d'un système pour l'utilisateur final, qui peut parfois correspondre à un besoin pressant (Lewis, 2010), est soit impossible car trop couteuse, soit au mieux loin d'être immédiate. Pourtant, le développement standard d'un tel système produit d'énormes bases de connaissances de traduction dont seule une infime partie est effectivement utilisée.

Des travaux précédents (Callison-Burch *et al.*, 2005; Lopez, 2008) permettent d'améliorer cette situation, en ne construisant que la partie utile des ressources de traduction qui sont nécessaires à la traduction d'un texte particulier. Dans ces approches, les tables de traduction sont construites *à la demande* pour chaque nouveau texte, en s'appuyant sur un échantillonnage *à taille constante des exemples de traduction* permettant l'obtention de ressources à la fois compactes et compétitives sur le plan de la qualité des traductions produites. Cependant, cet échantillonnage repose sur un pré-traitement qui est lui même très couteux. En particulier, l'alignement automatique au niveau des mots du corpus bilingue est supposé pré-exister, et son calcul peut requérir un temps considérable (par exemple, plus de 40 jours de calculs pour un processeur dans les expériences décrites Section 3).

Dans cet article, nous allons montrer qu'il est possible d'obtenir des performances très proches d'un système de référence à l'état de l'art, en effectuant à la demande non seulement la sélection des exemples de traduction, mais également l'alignement au niveau des mots des phrases parallèles correspondantes. Une telle approche est rendue possible par les travaux récents sur l'alignement ciblé de Gong *et al.* (2013), eux-mêmes fondés sur les résultats en alignement par échantillonnage et segmentation récursive de Lardilleux *et al.* (2012). En outre, le cadre proposé ici offre des perspectives originales au niveau du développement incrémental de systèmes de traduction adaptés (toute nouvelle phrase parallèle peut-être immédiatement utilisée), ainsi que de la fusion du développement des systèmes et de leur utilisation dans un cadre interactif (quelques secondes seulement sont nécessaires pour traduire une phrase sans autre ressource qu'un corpus parallèle et un modèle de langue cible).

	finally	,	what	our	fellow	citizens	are	demanding	is	the	right	to	information	.
enfin	**0.607**	0.001	ϵ	ϵ	0	ϵ	ϵ	0	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
,	0.001	**0.445**	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	0.001	ϵ	0.001	ϵ	0.001
c'	ϵ	ϵ	0.001	ϵ	ϵ	ϵ	ϵ	0	**0.036**	0.001	ϵ	ϵ	ϵ	ϵ
est	ϵ	ϵ	0.001	ϵ	ϵ	ϵ	ϵ	0	**0.223**	0.016	ϵ	0.001	ϵ	0.001
un	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	0.005	**ϵ**	ϵ	ϵ	ϵ	ϵ
droit	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	0	ϵ	ϵ	**0.084**	ϵ	ϵ	ϵ
à	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	0.001	ϵ	0.001	0.003	0.001	**0.018**	ϵ	ϵ
l'	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	0.002	0.009	ϵ	**0.002**	ϵ	ϵ
information	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	**0.499**	ϵ
que	ϵ	ϵ	0.002	ϵ	ϵ	ϵ	0.001	ϵ	0.002	0.001	ϵ	0.001	**ϵ**	ϵ
réclament	0	0	ϵ	ϵ	ϵ	ϵ	**ϵ**	**0.152**	ϵ	ϵ	0	0	0	ϵ
nos	ϵ	ϵ	**ϵ**	**0.171**	0.004	0.001	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
concitoyens	0	ϵ	ϵ	0.001	**0.323**	**0.009**	ϵ	ϵ	ϵ	ϵ	0	ϵ	0	ϵ
.	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	0.001	0.001	ϵ	ϵ	ϵ	**0.954**

FIGURE 1 – Illustration de la procédure d'alignement récursif (tiré de (Lardilleux *et al.*, 2012)). Les valeurs dans les cellules correspondent à des scores d'association ($0 < \epsilon \leq 0.001$). Les points d'alignement retenus par l'algorithme correspondent à ceux obtenus au niveau de récursion le plus profond (en gras).

Dans la section 2, nous décrivons le cadre proposé pour une construction efficace de systèmes reposant sur l'alignement de phrases à la demande (2.1) et l'estimation de modèles de traduction à la volée (2.2). Nous présentons dans la section 3 un ensemble d'expériences permettant de valider nos propositions : après la description du contexte expérimental (3.1), nous décrivons des configurations plus économes en temps dont les performances rejoignent, puis dépassent, celle d'un système de référence à l'état de l'art (3.2). Ces résultats sont discutés dans la section 4.

2 Développement efficace de systèmes de traduction automatique statistique

2.1 Alignement de phrases parallèles à la demande

Notre approche repose sur la capacité à aligner des phrases parallèles à la demande, afin d'en extraire des exemples de traduction pour des segments devant être traduits. Nous avons pour cela réutilisé l'approche de Gong *et al.* (2013), elle-même inspirée des travaux de Xu *et al.* (2006) et de Lardilleux *et al.* (2012), qui effectue un découpage binaire récursif des points d'alignement au niveau des mots possibles dans une paire de phrases parallèles (voir l'illustration Figure 1). Chaque point d'alignement entre paires de mots, correspondant à des cellules de la matrice d'alignement, est associé à un score d'association, obtenu ici par une procédure de ciblage de traduction par échantillonnage (voir (Gong *et al.*, 2013)). L'algorithme de segmentation commence au niveau d'un alignement complet entre les deux phrases, puis des points de coupe trouvés récursivement permettent de mettre en évidence des points d'alignement aux niveaux où plus aucune coupe n'est possible. Ce processus s'apparente à une analyse syntaxique descendante fondée sur les ITG (*Inversion Transduction Grammars*) (Wu, 1997), à la différence que le calcul de la segmentation s'effectue de façon gloutonne, en s'appuyant sur les scores d'association plutôt que sur un modèle probabiliste. Le travail de Gong *et al.* (2013) a montré que cette approche permettait d'obtenir des performances au niveau de l'état de l'art sur des tâches de traduction standard à petite échelle (avec alignement complet des corpus parallèles).

2.2 Estimation de modèles de traduction à la volée

Nous utilisons un tableau de suffixes (Manber & Myers, 1990) pour indexer la partie source des corpus, ce qui permet un accès rapide en $O(k)$ opérations pour des segments de k tokens. Un échantillon contenant des exemples d'un segment source à traduire $\bar{f}$ peut ainsi être obtenu efficacement et sa taille permet d'effectuer un compromis entre la précision de l'estimation et sa vitesse. Les approches précédentes utilisant cette technique (Callison-Burch *et al.*, 2005; Lopez, 2008) étaient fondées sur une échantillonnage aléatoire déterministe, garantissant que l'on choisissait toujours les mêmes exemples pour les mêmes segments. La probabilité de traduction d'un segment se calcule alors simplement par : $p(\bar{e}|\bar{f}) = count(\bar{f}, \bar{e}) / \sum_{\bar{e}'} count(\bar{f}, \bar{e}')$, avec $count(\cdot)$ le nombre d'occurrences d'un segment dans l'échantillon, incluant les occurrences où l'extraction d'une traduction est impossible (ce que Lopez (2008) décrit comme une estimation *cohérente*). L'échantillonage étant réalisé indépendamment pour chaque segment, il n'est pas possible d'estimer un modèle de traduction inverse, lequel peut néanmoins être approximé par : $p(\bar{f}|\bar{e}) = min(1.0, p(\bar{e}|\bar{f}) \times freq(\bar{f})/freq(\bar{e}))$, où $freq(\cdot)$ est la fréquence d'un segment dans le corpus entier. Les autres modèles utilisés sont les mêmes que dans la configuration par défaut d'un système `Moses` [1] fondé sur les segments (*phrase-based*).

1. `http://www.statmt.org/moses`

		# lignes	# tokens source	# tokens cible
apprentissage	WMT'13 + médical WMT'14	16,6M	396,9M	475,1M
développement	Cochrane	743	16,5K	21,4K
collection de test	Cochrane (100 documents)	1,8K	38,6K	49,3K

TABLE 1 – Description des corpus utilisés

3 Validation expérimentale

3.1 Contexte expérimental

Données Nous avons sélectionné la paire de langue anglais-français pour cette étude, car de très grandes quantités de données parallèles sont disponibles (plus de 400 millions de tokens anglais, soit plus de 15 fois la quantité utilisée dans l'étude de Lopez (2008)). Les données parallèles tout venant de l'atelier WMT ont été utilisées, ainsi que des données d'origines diverses du domaine médical [2]. Comme données d'évaluation, nous avons retenu les résumés systématiques de la littérature scientifique médicale produites par la collaboration Cochrane [3], qui présentent un certain nombre de caractéristiques souhaitables : les textes source en anglais sont de bonne qualité ; la langue d'origine des documents est toujours l'anglais, éliminant ainsi de possibles effets de traductions multiples ; les traductions de référence sont de bonne qualité, et la nature des documents tolère l'existence d'une seule traduction *normative* ; il est enfin possible de traiter ce corpus document par document. Le tableau 1 recense quelques statistiques utiles sur les différents corpus utilisés.

Systèmes Nous avons utilisé l'implémentation `mgiza++` [4] de l'aligneur statistique `giza++` (Och & Ney, 2003). Nous avons par ailleurs développé un aligneur à la volée tel que décrit section 2.1, en utilisant une taille d'échantillon de 100. Ce choix est motivé par les résultats en traduction fondée sur les segments donnés par Callison-Burch *et al.* (2005). Les heuristiques standard du système `Moses` pour l'extraction de traductions à partir de matrices d'alignement ont été utilisées, avec une taille de segment maximale de 6 tokens. Des modèles de langue 4-grammes pour le français ont été appris sur l'ensemble des données médicales en français de WMT'14. Le décodeur de `Moses` a été utilisé dans toutes les expériences et `KBMIRA` (Cherry & Foster, 2012) a été utilisé pour l'optimisation de certains systèmes.

Indicateurs de performance Nous avons utilisé les métriques standard BLEU (Papineni *et al.*, 2002) et TER (Snover *et al.*, 2006) et donnons des mesures correspondant à la moyenne des scores du test pour 3 optimisations indépendantes lorsqu'approprié. Comme indicateur principal du cout en ressources matérielles, le temps CPU utilisateur (*user CPU time*, tel que retourné par la commande `UNIX time`) a été utilisé comme focus sur la durée de calcul des processus mis en œuvre. Les programmes exécutés étant à différents niveaux de maturité et d'optimisation, ainsi qu'exécutés soit en code binaire (`mgiza++`) soit en dans une machine virtuelle Java (notre système), seules de larges différences pourront être considérées comme significatives.

3.2 Résultats expérimentaux

Cette section décrit les systèmes comparés dans nos expériences, dont les caractéristiques sont données dans la Table 2.

Système `moses` standard à très large échelle (`baseline-vanilla`) Afin de pouvoir comparer nos propositions à une approche bien connue et éprouvée, nous avons implémenté un système `Moses` utilisant l'intégralité des données. Le temps de développement correspond à l'exécution de `mgiza++` sur le corpus parallèle (1 006h), puis la construction des tables de traduction et de réordonnancement (206h), et enfin l'optimisation du système par `KBMIRA` (21h) après filtrage des tables pour les textes à traduire. Au total, 1 233h de temps processeur (plus de 51 jours) ont été nécessaires avant de pouvoir traduire la première phrase du corpus de test. En outre, les tables de traduction et de réordonnancement compressées occupent respectivement 20Go et 7,5Go, ce qui interdit leur installation sur la plupart des dispositifs mobiles.

2. Voir `http://www.statmt.org/wmt13` et `http://www.statmt.org/wmt14/medical-task`
3. `http://summaries.cochrane.org`
4. `http://www.kyloo.net/software/doku.php/mgiza:overview`

	BLEU	Δ	TER	Δ	temps CPU					
					TS	dev.	test	optim.	total	réduction
baseline-vanilla	34,12±0,10	-	48,59±0,22	-	-	1 212h		21h	1 233h	-
baseline-rnd	33,95±0,14	-0,17	48,79±0,13	+0,2	2h	1 006h		20h	1 028h	16%
config0	28,24	-5,88	49,92	+1,33	2h	0h	363h	0h	365h	70%
config1	28,87	-5,25	49,40	+0,81	2h	0h	111h	0h	113h	90%
config2	28,58	-5,54	49,54	+0,95	2h	0h	76h	0h	78h	93%
config2+spec	32,33	-1,79	46,42	-2,17	2h	0h	76h	0h	78h	93%
config2:tuned	33,38±0,06	-0,74	50,05±0,05	+1,46	2h	72h	76h	20h	170h	86%
config2:tuned+spec	37,63±0,05	+3,51	46,49±0,04	-2,10	2h	72h	76h	20h	170h	86%

TABLE 2 – Comparaison des performances de différents systèmes. Les résultats pour les systèmes optimisés (baseline, config2:tuned, config2:tuned+spec) sont décomposés en moyenne et écart type pour 3 optimisations.

Échantillonnage aléatoire des exemples de traduction (baseline-rnd) Nous avons également construit un système obtenu par sélection d'exemples d'apprentissage par échantillonnage aléatoire déterministe (Callison-Burch *et al.*, 2005; Lopez, 2008). Chaque segment d'un texte à traduire est cherché dans un tableau de suffixes du corpus source, et les alignements correspondant obtenus par mgiza++ sont utilisés pour extraire les traductions (ici encore, nous utilisons une taille d'échantillon de 100). Le temps de construction des tables de traduction et de réordonnancement est ici négligeable en regard du temps d'alignement, et le temps total avant traduction du corpus de test est de 1 028h (42 jours), une réduction de temps relativement modeste de 16% relativement à baseline-vanilla. Les gains en espace sont cependant bien évidemment beaucoup plus marquants, avec respectivement 32Mo et 8Mo pour les tables de traduction et de réordonnancement, lesquelles ne sont toutefois utiles que pour des documents particuliers. Finalement, la performance selon BLEU et TER est très proche de celle de baseline-vanilla. La réduction en temps de calcul et la construction à la demande des tables apparaissent déjà comme des modifications utiles.

Alignement des phrases à la demande, pas d'optimisation (config0) Nous considérons maintenant une configuration très simple : pour chaque document à traduire en séquence, les exemples de traduction sont choisis comme pour baseline-rnd et les phrases correspondantes sont alignées à la demande, puis les tables pour le document sont construites sur la base des alignements obtenus. Ces tables sont utilisées pour traduire chaque document en utilisant les paramètres par défaut du décodeur Moses. Le temps avant traduction du test (complet) n'inclut donc plus que le temps de construction du tableau de suffixes (2h) et l'estimation indépendante des tables pour l'ensemble des documents (363h), correspondant à 365h (15 jours), soit une réduction de temps sensible de 70% par rapport à baseline-vanilla. Cependant, le score BLEU se trouve très dégradé, avec une baisse de presque 6 points.

Mise en cache des tables de traduction et des alignements (config1) Nous avons implémenté un système de caches permettant de conserver (non sélectivement) d'une part les entrées des tables déjà construites pour des segments, et d'autre part les alignements de phrases parallèles déjà calculés par le système d'alignement à la demande. Si le premier type de cache n'apporte qu'un gain de temps marginal, le second s'avère plus performant, permettant de réduire le temps global à 113h (4,7 jours). Nous observons une légère amélioration dans les scores BLEU et TER.

Sélection d'exemples par réutilisation de phrases alignées (config2) Dans nos systèmes, le nombre de phrases à aligner à la demande a un impact direct sur le temps global. Nous avons donc étudié une technique simple qui réutilise préférentiellement des phrases déjà alignées. Ainsi, pour l'estimation des traductions d'un nouveau segment, les phrases déjà alignées qui le contiennent seront utilisées tout d'abord, puis éventuellement d'autres phrases seront alignées à la demande pour parvenir aux 100 exemples requis. Dans notre implémentation, l'échantillonnage aléatoire déterministe a d'abord lieu dans le sous-corpus déjà aligné, puis dans le corpus complémentaire. Cette technique mène à de faibles pertes en BLEU et TER par rapport à config1, mais permet pour un gain de temps sensible, désormais réduit à 78h (3,25 jours), soit une réduction de 93% par rapport à baseline-vanilla. Nous avons en outre mesuré une division par 5 du temps nécessaire (normalisé par la taille de chaque document) pour le calcul entre le premier et le dernier document de nos 100 documents de test, ce qui montre que notre approche parvient à exploiter très efficacement un cache qui grossit. Toutefois, l'écart en performance avec baseline-vanilla reste conséquent (environ 5,5 points BLEU).

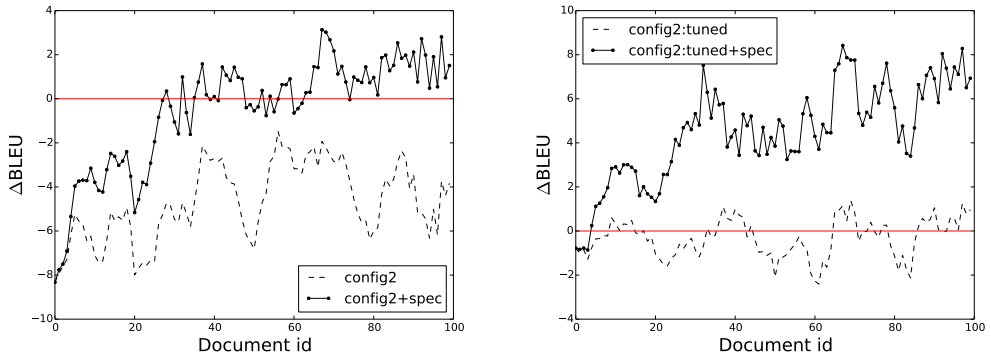


FIGURE 2 – Comparaison au niveau document relativement à `baseline-rnd`. En abscisses est représentée la séquence des documents traduits, et en ordonnées l'écart relatif en BLEU avec `baseline-rnd`.

Utilisation incrémentale de données du domaine (`config2+spec`) La configuration précédente est très en retrait par rapport à `baseline-vanilla`, cependant elle ne dépend pas de l'existence préalable d'un corpus de développement, lui-même très couteux à obtenir. Toutefois, notre aligneur à la demande nous permet trivialement d'ajouter aux données parallèles toute donnée nouvellement disponible. Nous considérons donc un scénario où chaque document complet devient immédiatement disponible après avoir été révisé. Nous alignons donc à la demande chaque document après sa traduction en utilisant sa traduction de référence (le faire pour l'ensemble des 99 premiers documents du test prend une demi-heure environ), puis utilisons les phrases parallèles alignées ainsi obtenues pour construire une nouvelle table de traduction contenant deux scores (modèle de traduction direct et pénalité de segment) dont les paramètres sont repris de la table principale (Hardt & Elming, 2010). Puisque cette table est estimée sur peu de données et qu'elle est de plus très sensible aux erreurs d'alignement, nous l'utilisons comme table secondaire (*back-off*) en complément de la table principale. Elle n'est donc utilisée que pour les longs segments non présents dans le grand corpus, ces segments correspondant possiblement à des termes et de la phraséologie propre à nos documents techniques.

Si le temps de développement reste sensiblement équivalent à `config2`, la performance en BLEU s'est très nettement rapprochée de celle de `baseline-vanilla`, puisque l'écart n'est plus que de 1,8 points. Il est intéressant de noter que cette configuration obtient un bien meilleur score TER que le système complet (baisse de 2,2 points), ce qui s'explique facilement par la réutilisation de grands segments spécifiques à la classe de documents et correspond donc à l'effet attendu d'une mémoire de traduction sous-phrastique. Il est particulièrement instructif de considérer comment la performance de traduction évolue pour chaque document i, qui utilise donc possiblement des exemples de traduction provenant des $i-1$ documents précédents. La partie gauche de la Figure 2 montre l'écart relatif en BLEU de nos deux variantes de `config2` par rapport à `baseline-vanilla`. Alors que, sans surprise, les documents sont systématiquement moins bien traduits par `config2`, `config2+spec` traduit moins bien les 30 premiers documents environ, puis traduit de mieux en mieux les documents. Ce résultat est particulièrement remarquable, puiqu'au bout d'environ 60 documents, `config2` traduit les documents systématiquement *mieux que le système de référence complet* ayant bénéficié d'une optimisation, tout en économisant 93% de temps de calcul.

Sélection d'exemples par réutilisation de phrases alignées et optimisation (`config2:tuned`) Nous nous ramenons maintenant à une situation plus usuelle, où un corpus de développement est utilisé pour optimiser le système ; la contrepartie en temps est la construction des tables pour la traduction de ce corpus, ramenant la réduction en temps par rapport à `baseline-vanilla` à 86% (soit 7 jours). La performance obtenue est alors très proche de celle de `baseline-vanilla`, avec une perte de seulement 0,7 points BLEU (métrique vers laquelle est effectuée l'optimisation). Ce résultat est là aussi particulièrement intéressant, car la performance obtenue est très proche de celle d'un système construit sur une très grande quantité de données.

Utilisation incrémentale de données du domaine et optimisation (`config2:tuned+spec`) L'étape naturelle suivante est de reprendre la configuration précédente `config2:tuned` et d'y adjoindre la possibilité étudiée précédemment d'exploiter de façon incrémentale un corpus additionnel constitué des documents précédemment traduits. Bien qu'il s'agisse ici d'une configuration sous-optimale (en particulier, parce que l'optimisation est faite une fois pour toutes et

ne prend pas en compte notre table additionnelle), les gains en traduction, toujours obtenus avec une réduction en temps d'environ 86% par rapport au système complet, sont relativement importants (+3,5 points BLEU et -2,1 points TER). On constate en outre (voir la partie droite de la Figure 2) une tendance à rapidement exploiter les nouvelles données, et donc à accroitre l'avantage vis-à-vis du système de référence tant que celui-ci n'a pas bénéficié d'un ré-entrainement (couteux) permettant d'ajouter les nouvelles données.

4 Discussion et perspectives

Nous avons décrit une série d'approches reposant sur le développement à la volée des ressources de systèmes de traduction statistique, ce qui permet un gain de temps considérable par rapport aux systèmes à l'état de l'art, tout en offrant, sous certaines conditions, des performances intéressantes en termes de qualité de traduction. Le résultat le plus marquant est la démonstration que des gains de traduction importants peuvent être obtenus lorsque les documents traduits sont immédiatement intégrés aux données d'apprentissage, ce qui permet même de se dispenser d'un corpus de développement. Il est ainsi possible d'aider un traducteur beaucoup plus rapidement que si celui-ci avait initialement à produire la traduction d'un corpus de développement.

L'exploitation incrémentale des données d'apprentissage pourrait aisément être menée au niveau de chaque phrase venant d'être révisée, laissant entrevoir des gains plus forts à l'intérieur des documents. Une autre perspective est la possibilité de réaliser une optimisation du système après chaque ajout significatif de données, plutôt qu'une fois pour toutes. Enfin, nous comptons étudier différentes manières d'améliorer encore les temps de calcul, en particulier en déterminant quels segments source ne sont pas utiles, voire correspondent à de mauvaises unités de traduction, ainsi qu'en stoppant l'échantillonnage des traductions pour un segment dès qu'une distribution de traductions stable est obtenue.

Références

CALLISON-BURCH C., BANNARD C. & SCHROEDER J. (2005). Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of ACL*, Ann Arbor, USA.

CHERRY C. & FOSTER G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of NAACL*, p. 427–436, Montréal, Canada.

GONG L., MAX A. & YVON F. (2013). Improving bilingual sub-sentential alignment by sampling-based transpotting. In *Proceedings of IWSLT*, Heidelberg, Germany.

HARDT D. & ELMING J. (2010). Incremental Re-training for Post-editing SMT. In *AMTA*, Denver, USA.

LARDILLEUX A., YVON F. & LEPAGE Y. (2012). Hierarchical sub-sentential alignment with Anymalign. In *Proceedings of the annual meeting of the European Association for Machine Translation*, p. 279–286, Trento, Italy.

LEWIS W. (2010). Haitian Creole : How to Build and Ship an MT Engine from Scratch in 4 days, 17 hours, & 30 minutes. In *Proceedings of EAMT*, Saint-Raphaël, France.

LOPEZ A. (2008). Tera-Scale Translation Models via Pattern Matching. In *Proceedings of COLING*, Manchester, UK.

MANBER U. & MYERS G. (1990). Suffix arrays : A new method for on-line string searches. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, p. 319–327.

OCH F. J. & NEY H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, **29**(1), 19–51.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). BLEU : a method for automatic evaluation of machine translation. In *Proceedings of ACL*, p. 311–318.

SMITH J. R., SAINT-AMAND H., PLAMADA M., KOEHN P., CALLISON-BURCH C. & LOPEZ A. (2013). Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of ACL*, Sofia, Bulgaria.

SNOVER M., DORR B., SCHWARTZ R., MICCIULLA L. & MAKHOUL J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*, p. 223–231, Cambridge, USA.

WU D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, **23**(3), 377–404.

XU J., ZENS R. & NEY H. (2006). Partitioning parallel documents using binary segmentation. In *Proceedings of WMT*, p. 78–85.

On-going Cooperative Research towards Developing Economy-Oriented Chinese-French SMT Systems with a New SMT Framework

Yidong CHEN[1], Lingxiao WANG[2], Christian BOITET[2], Xiaodong SHI[1]
(1) School of Information Science and Technology, Xiamen University, Xiamen, Fujian, China
(2) GETALP, Laboratoire d'Informatique Grenoble (LIG), Université Joseph Fourier, Grenoble, France
ydchen@xmu.edu.cn

Résumé. Nous présentons un projet collaboratif en cours mené par l'université de Grenoble et l'université de Xiamen, et visant à créer des instances d'un nouveau type de système de traduction automatique statistique utilisant des ressources lexico-sémantiques et discursives. Le but concret est de développer des systèmes de TAS chinois-français pour des sites boursiers et économiques. Comme très peu de corpus et de dictionnaires bilingues chinois-français sont disponibles sur Internet, l'anglais est utilisé comme "pivot" pour construire les équivalents chinois-français par transitivité. Outre la description générale de ce projet, nous décrivons les progrès sur deux axes de recherche liés à ce projet. Pour cela, nous utilisons une méthode, proposée par XMU, d'induction de probabilité fondée sur la similarité thématique, qui produit des tables de traduction C-F à partir de tables de traduction C-E et E-F. Pour disposer de bons corpus parallèles C-F, nous utilisons un système Web de post-édition collaborative qui peut déclencher l'amélioration incrémentale du système de TA en utilisant des métriques d'évaluation de TA et en extrayant la "meilleure partie" de la mémoire de traductions courante.

Abstract. We present an on-going collaborative project pursued by Grenoble University and Xiamen University and aiming at creating instances of a new kind of SMT system using semantics and discourse-related resources. The concrete goal is to develop Chinese-French systems specialized to stock option and economic websites. Since very few Chinese-French bilingual corpora and dictionaries are freely available on Internet, English is used as a "pivot" for constructing the Chinese-French translation equivalents by transitivity. For this, we use a method, proposed by XMU, of probability induction based on topic similarity, which produces C-F translation tables from C-E and E-F translation tables. For getting good C-F parallel corpora, we use a web-based collaborative post-editing system that can trigger the incremental improvement of the MT system by using MT evaluation metrics and extracting the "best part" of the current translation memory.

Mots-clés : traduction automatique statistique (SMT), chinois-français, domaine économique
Keywords: SMT, Chinese-French, Economic Domain

1 Introduction

The desire and need for cross-cultural communication between China and Europe, both officially and non-governmentally, are on the increase. Especially, France is an important strategic partner of China and has deep relationships with China in many fields such as economy, science and technology, culture and education, etc. But the language barrier between Chinese and French is impassable in most situations.

Concerning the economy, the cooperation between China and France grew rapidly in the past decades. France is now China's fourth largest trade partner in the EU, behind Germany, the Netherlands and the UK. According to data from the National Bureau of Statistics of China, bilateral trade between China and France increased from $13.4 billion in 2003 to $51 billion in 2012. Two-way direct investments are also on the rise. With the continuous improvement of China's investment environment, more and more French investors intend to invest in China. However, most portals of China, such as the website of Shanghai Stock Exchange and the website of Shenzhen Stock Exchange, only provide a Chinese version and an English version, but no French version. Moreover, Chinese investors are also more and more on the look for opportunities in France and other francophone areas, notably in Africa.

Although French-Chinese bilingual applications are urgently needed in many situations, not much work has been done yet on French-Chinese MT, and French-Chinese is still an under-resourced language pair as there are no large good

quality freely available bilingual lexico-semantic resources, and no sufficiently good MT systems. Most French-Chinese MT systems, in particular GT (GoogleTranslate[1]) use English text as a pivot, which introduces more errors, often degrading translation quality to uselessness, as illustrated below. Results may be useful for guessing the topic of a text, but are otherwise bad or misleading, while investors need precise translations to decide whether to invest and where. Here are two examples, both selected from the announcements of the Shanghai Stock Exchange. We show the result of GT: output of GT-zh-en and then of en-fr (calling GT on zh-fr gives the same outputs) and en-PE-fr (post-edited en).

Ex1 (Source): 恢复交易后，如该证券交易中再次出现异常情况，本所可实施第二次停牌，停牌时间持续至今日收盘前五分钟。
(GT-zh-en) After the resumption of trading in the securities trading as abnormal situation occurs again, this can be implemented by the second suspension, suspension lasted until today the closing five minutes.
(GT-zh-en-PE) If the trading of this security appears to be abnormal again after resumption of its trading, we may perform a second suspension upon it and this suspension will last until five minutes before today's closing.
(GT-zh-en-fr = GT-zh-fr): Après la reprise de la négociation dans le négoce de titres comme situation anormale se produit de nouveau, ce qui peut être mis en œuvre par la deuxième suspension, la suspension **a duré** jusqu'à aujourd'hui les cinq dernières minutes.
(GT-zh-en-PE-fr): Si la négociation de ce titre semble être anormal à nouveau après la reprise de ses opérations, nous pouvons effectuer une deuxième suspension sur elle et cette suspension durera jusqu'à cinq minutes avant la clôture d'aujourd'hui.

Ex2 (Source): 四、凡 2013 年 12 月 31 日前在本所上市的公司债券，以及 2014 年在本所上市时未披露 2013 年年度报告的公司债券，应于 2014 年 4 月 30 日前完成本次年度报告的披露工作。
(GT-zh-en) Fourth, where the corporate bond December 31, 2013 are listed in this, as well as the 2014 listed in the 2013 annual report of undisclosed corporate bonds should be April 30, 2014 disclosed in the annual report is completed work.
(GT-zh-en-PE) Fourth, the bonds which were listed before December 31, 2013 or listed in 2014, but have not yet disclosed their annual reports of 2013, should complete the disclosure of their annual reports before April 30, 2014.
(GT-zh-en-fr = GT-zh-fr): Quatrièmement, lorsque **le lien** de l'entreprise 31 Décembre , 2013 **figurent dans cet , ainsi que de 2014** figurant dans le rapport annuel 2013 d'obligations de sociétés non divulgués devrait être de 30 Avril , 2014 communiquées dans le rapport annuel **est terminé travail**.
(GT-zh-en-PE-fr): Quatrièmement, les **liens** qui ont été répertoriés avant le 31 Décembre 2013 répertoriés en 2014, mais n'ont pas encore divulgué leurs rapports annuels de 2013, devrait compléter la divulgation de leurs rapports annuels avant le 30 Avril 2014.

Examples above show that GT uses English as a pivot for the zh-fr direction. However, it is not the case for the fr-zh direction: outputs are different. Google Translate is claimed to be one of the "state-of-the-art" MT system. The truth is however that "state-of-the-art" for general MT systems means "very bad to useless": not knowing Chinese, one is at a loss to guess the exact meaning. Also, even if an output looks good (due to the use of a good target language model), its reliability may be very low, especially when negation appears or disappears at random (ex. 2 in French), or, as in these examples, when essential words are badly translated ("If" → "après" [after] and not "si", "bond" → "lien" [link] and not "obligation", "last" → "dernière" instead of "durer"), grammar is wrong leading to ununderstandability (ex. 2 in French, at several points), dependencies have been modified ("its resumption of" should be "resumption of its"), and bad handling of time in English leads to incorrect tenses in French (ex. 1 and 2). Actually, Example 2 is total gibberish in French. No sense can be extracted from it. Example 1 has actually a *negative* adequacy[2] in French, as it gives counterfactual and misleading information (the suspension "has lasted until…" instead of "will last until…").

This illustrates the need of building direct MT systems between French and Chinese, specialized to the domains (stock option markets, economy) and to the corresponding sublanguages. That will certainly considerably improve MT quality. But, even if one augments the BLEU by 25% or more[3], the problems above remain. In situations where exactness of meaning is crucial and post-editing would require professionals working round the clock (because flash reports have a life expectancy of only a few hours). To improve quality, semantic and discourse information should be used.

Section 2 will now give an overall description of this project. Then two aspects where progress has been made will be described in Section 3 in more detail.

[1] http://translate.google.com/

[2] Adequacy is usually measured on a scale from 0 to 5, but should rather be similar to a kappa coefficient, ranging from -1 to +1.

[3] In an experiment conducted at the EU, a Moses system was built on 50,000 sentences of the corpus of the Council. Compared to the unspecialized system built on 20 M sentences (400 times more!), it showed an *increase by 25%* of the BLEU measure.

2 Overview of the Project

As mentioned in Section 1, the overall goal of this project is to construct economy-oriented Chinese-French computational linguistic resources, e.g. bilingual semantic resources, parallel corpora, and discourse structure annotation banks, etc., then propose a new SMT model making use of semantic and discourse information, and finally build a web-based collaborative post-editing platform. We distinguish three levels: data development, fundamental research, and tool building. FIGURE 1 shows the overall organization of this project, in which three points should be noted.

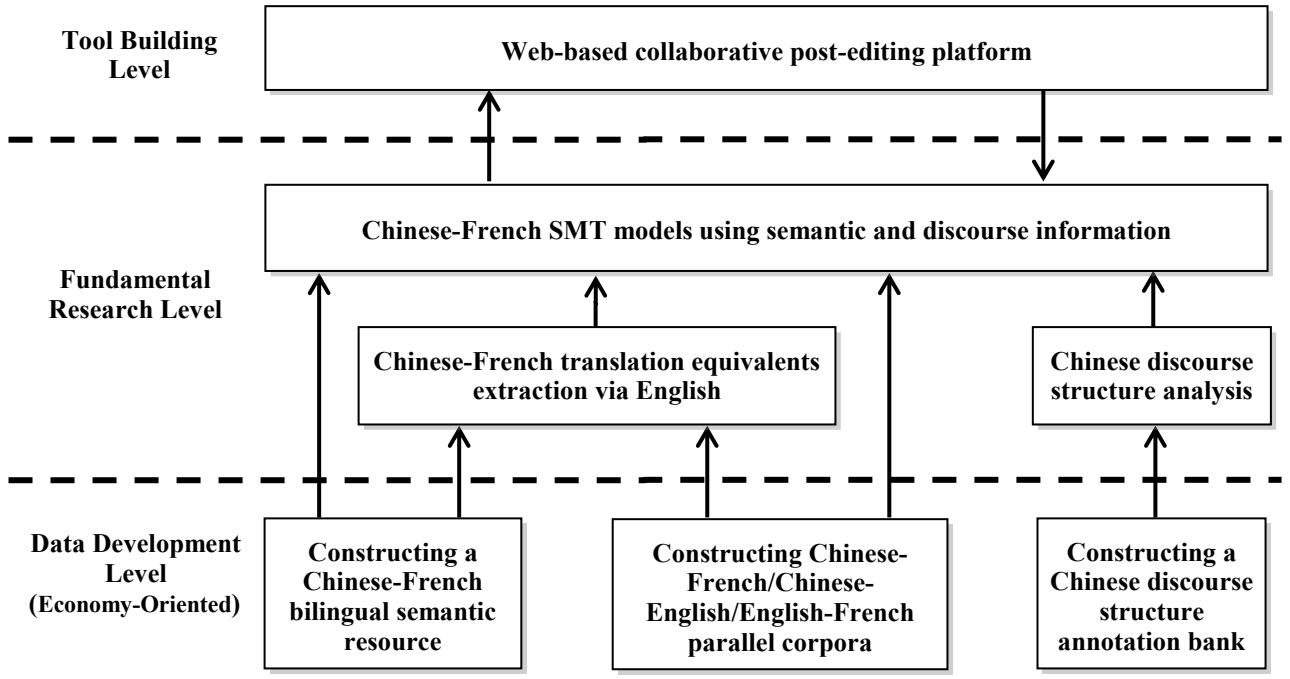


FIGURE 1 : Overall picture of the project organization

Firstly, a pivot-based method is used for extracting the Chinese-French translation equivalents. We choose to use English as a pivot, because it is much easier to gather large-scale Chinese-English and English-French parallel texts than to directly collect Chinese-French ones. Moreover, pivot-based methods (Wu and Wang, 2007; Paul et al., 2009; Wu and Wang, 2009) have been proven to be effective when building SMT systems for under-resourced language pairs.

Secondly, the translation model uses discourse-level information. Actually, recent research has shown that discourse-level information is quite important for machine translation systems, especially those concerning Chinese (Gong et al., 2011, 2012; Tu et al., 2013). As Chinese does not have grammatical markers of tense, the time at which an action takes place usually has to be inferred from the context. Consider again example 1 in Section 1. The source sentence contains three clauses, namely "恢复交易后，如该证券交易中再次出现异常情况 (If the trading of this security appears to be abnormal again after resumption of its trading,)", "本所可实施第二次停牌 (we may perform a second suspension upon it)" and "停牌时间持续至今日收盘前五分钟 (and this suspension will last until five minutes before today's closing.)". Because of the omission of the object in the second clause, we don't know what the second suspension will be executed for, and the unclearness of the tense in the third clause also contributed to the inadequacy of the translation. Suppose now that we have the discourse graph shown in FIGURE 2 below. Both the omitted information and the tense information could then be inferred. Therefore, we decide to conduct research on Chinese-French MT making use of discourse-level information.

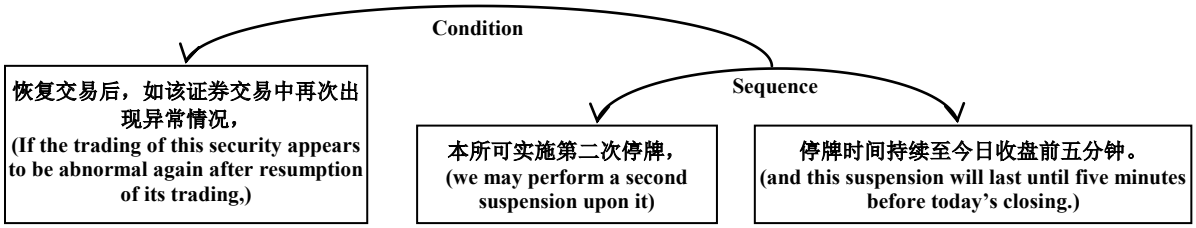


FIGURE 2 : The discourse graph of the source sentence of Example 1

Thirdly, semantic knowledge is used when extracting the translation equivalents. We would do so because we found that the previous pivot-based approaches, which do not take semantic knowledge into account, may produce incorrect translation equivalents due to the lack of sufficient context. For example, suppose we have a phrase pair "银行 ↔ bank" in the Chinese-English bilingual phrase table, and phrase pairs "bank ↔ la banque" and "bank ↔ la rive" in the English-French bilingual phrase table. Then, using the transfer method (Paul et al., 2009; Wu and Wang, 2009), we get two Chinese-French phrase pairs, i.e. "银行 ↔ la banque" and "银行 ↔ la rive". However, the latter is obviously wrong. We expect that, by incorporating semantic information, this problem could be overcome.

We distinguish seven subtasks in this project (see FIGURE 1), briefly described below.

1) Constructing Chinese-French bilingual lexico-semantic resources

This subtask aims at creating bilingual lexico-semantic data by enriching HowNet (Dong and Dong, 2006), a Chinese-English conceptual database, with French data. In this subtask, economy-related bilingual terms will be integrated into the semantic resource.

2) Constructing Chinese-French/Chinese-English/English-French parallel corpora

In order to carry out SMT research, parallel corpora are needed. Since very few Chinese and French bilingual data are freely available on Internet, it is not easy to construct large-scale Chinese-French parallel corpora, especially economy-oriented ones. Therefore, it is reasonable to build the resources of Chinese-French SMT systems using English as a pivot. To this end, we are building Chinese-English and English-French bilingual corpora related to the domain of economy. Then, to support learning structure transformation knowledge between Chinese and French, a medium-scale Chinese-French parallel corpus will also be created.

3) Constructing Chinese discourse structure annotation bank

In order to incorporate discourse-level information in our Chinese-French SMT system, we first need a Chinese discourse structure analysis module trained on Chinese discourse structure annotation banks. However, the Chinese discourse structure annotation banks, currently under construction to support discourse-based Chinese-related MT research, are not yet available now (Li et al., 2012). Hence, this subtask aims at constructing a Chinese discourse structure annotation bank based on the Chinese part of the economy-oriented parallel corpora constructed in Subtask 2. The course of constructing the annotation bank could be summarized as two steps. Firstly, sentences are partitioned into sentence groups according to their topic similarities. Secondly, the relationships among the sentences in the same groups, as well as the time information of the sentences or the time relationships between sentence pairs are humanly tagged. The annotation result for example 1 in Section 1 may look like as follows:

(Discourse (Sentences (S1, 0-21), (S2, 22-32), (S3, 33-48)),
(Sentence-relationships (Condition S1, S2), (Sequence S2, S3)),
(Time-information Time(S1)=present, Time(S2)>Time(S1), Time(S3)=Time(S2)))

4) Chinese-French translation equivalents construction via English

Given the constructed Chinese-English and English-French parallel corpora, the objective of the subtask is to construct Chinese-French translation equivalents. In the course of equivalents extraction, semantic information based on the lexico-semantic resources constructed in Subtask 1 will be considered.

5) Chinese discourse structure analysis

The objective of this subtask is to research and propose a Chinese discourse structure analysis model based on the Chinese discourse structure annotation bank constructed in Subtask 3.

6) Chinese-French SMT models using semantic and discourse information

The goal of this subtask is to propose a new Chinese-French SMT model using semantic and discourse information. In the course of translation, the tense and semantic consistency across sentences will be considered, based on the discourse graphs produced by the analysing module in Subtask 5 and the time model trained with the annotation bank in Subtask 3.

7) Web-based collaborative post-editing platform

The objective of this subtask is to build a collaborative web-based post-editing platform. This platform is built based on iMAG/SECTra (Wang and Boitet, 2013), and incorporates the economy-oriented SMT system developed in Subtask 6.

3 Some preliminary progresses

This project started in September 2013. Since then, we progressed on two fronts: data collection and pivot-base construction of bilingual equivalents.

1) Data collection

From websites related to stock exchange, about 1000 bilingual web pages have been crawled and handled so far (Table 1 shows the statistics of this dataset).

Direction	# of pages	Size
zh-en	761	39.4M
en-fr	250	13.5M

TABLE 1: Statistics of the data collected so far

At the same time, with the participation of 2 Chinese students, we started the process of creating a Chinese-French parallel corpus via post-editing, using an iMAG collaborative gateway, as in (Wang and Boitet, 2013).

2) Research on the pivot-based construction of bilingual equivalents

In previous pivot-based methods, such as (Wu and Wang, 2007), the translation probability induction may become inaccurate if the semantic tendencies of the source-pivot (SP) corpus and the pivot-target (PT) corpus are different. To overcome this problem, an effective method is to use context information to measure the semantic similarity of the rule pairs. To solve this problem, we have proposed and validated a pivot probability induction method based on topic similarity information. It consists of two steps.

Firstly, we use the topic model (Blei, 2003) to discover the latent topic structure for the pivot language document and each synchronous pivot phrase rule is then attached with the topic distribution according to the document it comes from. Formula 1 is used to assign topic distribution to a given rule.

$$P(z_i \mid r) = \frac{\sum_{D \in \sum_*} count(I, D) P(z_i \mid d_p)}{\sum_{z \in Z} \sum_{D \in \sum_*} count(I, D) P(z \mid d_p)} \quad (1)$$

where D is a bilingual document in the given bilingual corpus $\sum_*$. $P(z_i \mid d_p)$ represents the probability of the ith topic of the pivot language document d_p in D. In addition, the function $count(I, D)$ denotes the frequency of the rule instance I in D.

Secondly, the probability induction is executed by computing the topic similarity for the rule pairs instead of using simple phrase translation probability multiplication between the SP and PT translation models. Formulas 2-3 are used:

$$\phi(\bar{t} \mid \bar{s}) = \frac{\sum_{\bar{p}} sim(P(Z \mid r_{\bar{s}\bar{p}}), P(Z \mid r_{\bar{p}\bar{t}}))}{\sum_{\bar{t}} \sum_{\bar{p}} sim(P(Z \mid r_{\bar{s}\bar{p}}), P(Z \mid r_{\bar{p}\bar{t}}))} \quad (2)$$

$$\phi(\bar{s} \mid \bar{t}) = \frac{\sum_{\bar{p}} sim(P(Z \mid r_{\bar{t}\bar{p}}), P(Z \mid r_{\bar{p}\bar{s}}))}{\sum_{\bar{s}} \sum_{\bar{p}} sim(P(Z \mid r_{\bar{t}\bar{p}}), P(Z \mid r_{\bar{p}\bar{s}}))} \quad (3)$$

where $\bar{s}$, $\bar{p}$, $\bar{t}$ and Z are the source phrase, pivot phrase, target phrase and topic distribution, respectively. $sim(x, y)$ is the similarity function, and is defined as the cosine (pseudo-distance) of vectors x and y (Formula 4).

$$sim(x, y) = \cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \|y\|} \quad (4)$$

The experimental results showed that our method greatly outperforms the baseline. A paper related to this work was published in the *Journal of Computational Information Systems* in 2013 (Huang et al., 2013).

4 Conclusion

This paper gives an introduction of a joint project between GETALP and the NLP lab of XMU on building economy-oriented Chinese-French SMT systems, as well as its preliminary progress. We demonstrate the need of building direct Chinese-French MT systems specialized field to that field and its sublanguages, and also explain why it seems to be necessary to incorporate semantic and discourse-based information to obtain a quality (of raw MT) sufficient for the needs of users, in a situation when usually there is no time for post-editing. This project is still in its initial stages and many efforts are required in the future. However, specialized versions not yet using the new semantic- and discourse-related features should be demonstrable at the time of the conference. We hope and believe that the future results of this project will be useful in overcoming the language barrier of the communications between China and France in the economy-related field.

Acknowledgements

This work was supported by the State Scholarship Fund of China (Grant No. 201208350055), the National Natural Science Foundation of China (Grant No. 61005052) and the Natural Science Foundation of Fujian Province of China (Grant No. 2011J01369). It was also partly supported by the French ANRT agency and the Lingua et Machina firm.

References

BLEI D. M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning 3*, 993-1022.

DONG, Z.,AND DONG, Q. (2006). HowNet and the Computation of Meaning. *World Scientific Publishing Co.Pte.Ltd*.

GONG, Z., ZHANG, M., ZHOU, G. (2011). Cache-based Document-level Statistical Machine Translation. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, Edinburgh, UK, pp. 909-919.

GONG, Z., ZHANG, M., TAN, C., ZHOU, G. (2012). N-gram-based Tense Models for Statistical Machine Translation. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju, Korea, pp. 276-285.

HUANG, Y., SHI, X., SU, J., CHEN Y., HUANG, G. (2013). Pivot Probability Induction for Statistical Machine Translation with Topic Similarity. *Journal of Computational Information Systems*, 20(9), 8351-8359.

LI, Y., ZHU, K., ZHOU, G. (2012). Summary of research on English discourse parsing. *Application Research of Computers*, 29(6).

PAUL, M., YAMAMOTO, H., SUMITA, E., AND NAKAMURA S. (2009). On the Importance of Pivot Language Selection for Statistical Machine Translation. *Proceedings of NAACL-HLT'09*, 2009, pp. 221-224.

TU, M., ZHOU Y., AND ZONG C. (2013). A Novel Translation Framework Based on Rhetorical Structure Theory. *The 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, Sofia, Bulgaria.

WANG, L., AND BOITET, C. (2013), Online production of HQ parallel corpora and permanent task-based evaluation of multiple MT systems: both can be obtained through iMAGs with no added cost. *Proceedings of MT Summit XIV, The 2nd Workshop on Post-Editing Technologies and Practice.* Nice, France 2 - 6 September 2013.

WU, H. AND WANG H. (2007). Pivot Language Approach for Phrase-Based Statistical Machine Translation. *Proceedings of ACL'07*, 2007, pp. 856-863.

WU, H. AND WANG, H. (2009). Revisiting Pivot Language Approach for Machine Translation. *Proceeding of ACL'09*, 2009, pp. 154-162.

Extraction automatique de termes combinant différentes informations

Juan Antonio Lossio-Ventura[1] Clement Jonquet[1] Mathieu Roche[1,2] Maguelonne Teisseire[1,2]
(1) LIRMM, Université de Montpellier 2, CNRS, Montpellier - France
(2) Irstea, Cirad, TETIS, Montpellier - France
juan.lossio@lirmm.fr, clement.jonquet@lirmm.fr, mathieu.roche@cirad.fr,
teisseire@teledetection.fr

Résumé. Pour une communauté, la terminologie est essentielle car elle permet de décrire, échanger et récupérer les données. Dans de nombreux domaines, l'explosion du volume des données textuelles nécessite de recourir à une automatisation du processus d'extraction de la terminologie, voire son enrichissement. L'extraction automatique de termes peut s'appuyer sur des approches de traitement du langage naturel. Des méthodes prenant en compte les aspects linguistiques et statistiques proposées dans la littérature, résolvent quelques problèmes liés à l'extraction de termes tels que la faible fréquence, la complexité d'extraction de termes de plusieurs mots, ou l'effort humain pour valider les termes candidats. Dans ce contexte, nous proposons deux nouvelles mesures pour l'extraction et le "ranking" des termes formés de plusieurs mots à partir des corpus spécifiques d'un domaine. En outre, nous montrons comment l'utilisation du Web pour évaluer l'importance d'un terme candidat permet d'améliorer les résultats en terme de précision. Ces expérimentations sont réalisées sur le corpus biomédical GENIA en utilisant des mesures de la littérature telles que *C-value*.

Abstract. Comprehensive terminology is essential for a community to describe, exchange, and retrieve data. In multiple domain, the explosion of text data produced has reached a level for which automatic terminology extraction and enrichment is mandatory. Automatic Term Extraction (or Recognition) methods use natural language processing to do so. Methods featuring linguistic and statistical aspects as often proposed in the literature, rely some problems related to term extraction as low frequency, complexity of the multi-word term extraction, human effort to validate candidate terms. In contrast, we present two new measures for extracting and ranking muli-word terms from domain-specific corpora, covering the all mentioned problems. In addition we demonstrate how the use of the Web to evaluate the significance of a multi-word term candidate, helps us to outperform precision results obtain on the biomedical GENIA corpus with previous reported measures such as *C-value*.

Mots-clés : Extraction Automatique de Termes, Mesure basée sur le Web, Mesure Linguistique, Mesure Statistique, Traitement Automatique du Langage Biomédical.

Keywords: Automatic Term Extraction, Web-based measure, Linguistic-based measure, Statistic-based measure, Biomedical Natural Language Processing.

1 Introduction

Les méthodes d'Extraction Automatique de Termes (EAT) visent à extraire automatiquement des termes techniques à partir d'un corpus. Ces méthodes sont essentielles pour l'acquisition des connaissances d'un domaine pour des tâches telles que la mise à jour de lexique. En effet, les termes techniques sont importants pour mieux comprendre le contenu d'un domaine. Ces termes peuvent être : (i) composés d'un seul mot (généralement simple à extraire), ou (ii) composés de plusieurs mots (difficile à extraire). Notre travail concerne plus spécifiquement l'extraction de termes composés de plusieurs mots.

Les méthodes d'EAT impliquent généralement deux étapes principales. La première extrait des candidats en calculant "l'unithood" qui qualifie une chaîne de mots comme une expression valide (Korkontzelos *et al.*, 2008). La deuxième étape calcule le "termhood" qui sert à mesurer la spécificité propre à un domaine.

Il existe quelques problèmes connus de l'EAT tels que : (i) l'extraction de termes non pertinents (bruit) ou le nombre réduit de termes pertinents retournés (silence), (ii) l'extraction de termes de plusieurs mots qui ont inévitablement des structures complexes, (iii) l'effort humain dans la validation manuelle des termes candidats, (iv) l'application aux corpus

de grande échelle. En réponse à ces problèmes, nous proposons deux nouvelles mesures. La première, appelée *LIDF-value*, est fondée sur l'information statistique et linguistique. La mesure *LIDF-value* permet de mieux prendre en compte l'unithood, en lui adossant un niveau de qualité. Elle traite les problèmes i), ii) et iv). La seconde, appelée *WAHI*, est une mesure fondée sur le Web traitant les problèmes i), ii) et iii). Dans cet article, nous comparons la qualité des méthodes proposées avec les mesures de référence les plus utilisées. Nous démontrons que l'utilisation de ces deux mesures améliore l'extraction automatique de termes spécifiques d'un domaine, à partir de textes qui n'offrent pas une fiabilité statistique liée aux fréquences.

Le reste du papier est organisé comme suit : nous discutons tout d'abord des travaux connexes dans la Section 2. Les deux nouvelles mesures sont ensuite décrites en Section 3. L'évaluation en termes de précision est présentée dans la Section 4 suivie des conclusions en Section 5.

2 État de l'art

Plusieurs études récentes se sont concentrées sur l'extraction des termes de plusieurs mots (n-grammes) et d'un seul mot (unigrammes). Les méthodes d'extraction de terme existantes peuvent être divisées en quatre grandes catégories : (i) *linguistique*, (ii) *statistique*, (iii) *apprentissage automatique*, et (iv) *hybride*. La plupart de ces techniques appartiennent à des approches de fouille de textes. Les techniques existantes fondées sur le Web ont rarement été appliquées à l'EAT, mais comme nous le verrons, ces approches peuvent être adaptées à cet objectif.

Méthodes de Fouille de Textes

Les méthodes liées à la fouille de textes combinent en général différents types d'approches : linguistique (Gaizauskas *et al.*, 2000; Krauthammer & Nenadic, 2004), statistiques (Van Eck *et al.*, 2010) ou fondées sur un apprentissage automatique pour l'extraction et la classification des termes (Newman *et al.*, 2012). Il faut également citer les propositions issues du domaine de l'Extraction Automatique de Mots Clés (EAMC) dont les mesures peuvent être adaptées pour l'extraction de termes d'un corpus (Lossio-Ventura *et al.*, 2013) (Lossio-Ventura *et al.*, 2014).

Les méthodes hybrides sont principalement linguistiques et statistiques. *GlossEx* (Kozakov *et al.*, 2004) estime la probabilité du mot dans un corpus de domaine comparée à la probabilité du même mot dans un corpus général. *Weirdness* (Ahmad *et al.*, 1999) estime que la distribution des mots dans un corpus spécifique est différente de la distribution des mots dans un corpus général. *C/NC-value* (Frantzi *et al.*, 2000), qui combine l'information statistique et linguistique pour l'extraction de termes de plusieurs mots et des termes imbriqués, est la mesure la plus connue dans le domaine biomédical. Dans (Zhang *et al.*, 2008), les auteurs montrent que *C-value* a des meilleurs résultats par rapport à d'autres mesures citées ci-dessus. Une autre mesure est *F-TFIDF-C* (Lossio-Ventura *et al.*, 2014) qui combine une mesure d'EAT *(C-value)* et une mesure d'EAMC *(TF-IDF)* pour extraire des termes obtenant des résultats plus satisfaisants que *C-value*. Aussi, *C-value* a été appliquée à de nombreuses langues autres que l'anglais, comme le japonais, serbe, slovène, polonais, chinois, espagnol, arabe, et français (Ji *et al.*, 2007; Barron-Cedeno *et al.*, 2009; Lossio-Ventura *et al.*, 2013). Ainsi, nous proposons d'adopter *C-value* et *F-TFIDF-C* comme référence pour l'étude comparative au cours de nos expérimentations.

Méthodes de Fouille du Web

Différentes études de fouille du Web se concentrent sur la similarité et les relations sémantiques. Les mesures d'association de mots peuvent être divisées en trois catégories (Chaudhari *et al.*, 2011) : (i) *Co-occurrence* qui s'appuient sur les fréquences de co-occurrence de deux mots dans un corpus, (ii) *Basées sur la similarité distributionnelle* qui caractérisent un mot par la distribution d'autres mots qui l'entourent, et (iii) *Basées sur les connaissances*, comme les thésaurus, les réseaux sémantiques, ou taxonomies. Dans cet article, nous nous concentrons sur les mesures de co-occurrence, car notre objectif est d'extraire les termes de plusieurs mots et nous proposons de calculer un degré d'association entre les mots qui composent un terme. Les mesures d'association de mots sont utilisées dans plusieurs domaines comme l'écologie, la psychologie, la médecine et le traitement du langage. De telles mesures ont été récemment étudiées dans (Pantel *et al.*, 2009) (Zadeh & Goel, 2013), telles que *Dice*, *Jaccard*, *Overlap*, *Cosine*. Une autre mesure pour calculer l'association entre les mots utilisant les résultats des moteurs de recherche Web est la Normalized Google Distance (Cilibrasi & Vitanyi, 2007). Elle s'appuie sur le nombre de fois où les mots apparaissent ensemble dans le document indexé par un moteur de recherche. Dans cette étude, nous comparons les résultats de notre mesure fondée sur le Web avec les mesures d'association de référence (*Dice*, *Jaccard*, *Overlap*, *Cosine*).

3 Deux nouvelles mesures pour l'extraction de termes

3.1 Une mesure fondée sur l'information linguistique et statistique : *LIDF-value*

Notre première contribution consiste à donner une meilleure importance à l'unithood des termes afin de détecter des termes avec une faible fréquence.

De manière similaire aux travaux de la littérature, nous supposons que les termes d'un domaine ont une structure syntaxique similaire. Par conséquent, nous construisons une liste de patrons linguistiques les plus courants selon la structure syntaxique des termes techniques présents dans un dictionnaire. Dans notre cas, il s'agit d'UMLS [1] qui est un ensemble de référence de terminologies biomédicales. Dans un premier temps, nous effectuons l'étiquetage grammatical des termes contenus dans le dictionnaire en utilisant Stanford CoreNLP API (POS tagging) [2]. Ensuite nous calculons la fréquence des structures syntaxiques. Nous sélectionnons les 200 fréquences les plus élevées pour construire la liste de patrons (ou motifs) qui seront pris en considération. Cette liste est pondérée selon la fréquence d'apparition de chaque patron par rapport à l'ensemble des motifs. Un terme candidat est alors retenu s'il appartient à la liste des structures syntaxiques sélectionnées. Le nombre de termes utilisés pour construire cette liste était de 2 300 000. La Figure 1 illustre le calcul de la *probabilité* des patrons linguistiques.

Pattern	Frequency	Probability
NN IN JJ NN IN JJ NN	3006	3006/4113 = 0,73
NN CD NN NN NN	1107	1107/4113 = 0,27
	4113	1,00

FIGURE 1: Exemple de construction de patrons linguistiques (où *NN* : nom, *IN* : préposition, *JJ* : adjectif, et *CD* : numéro).

L'objectif de notre mesure, *LIDF-value* (***L**inguisitic patterns*, ***IDF***, and ***C-value*** information) est de calculer le termhood pour chaque terme, en utilisant la *probabilité* calculée précédemment, également avec l'idf, et *C-value* de chaque terme. La fréquence inverse de document (idf) est une mesure indiquant si le terme est commun ou rare dans tous les documents. Il est obtenu en divisant le nombre total de documents par le nombre de documents contenant le terme, puis en prenant le logarithme de ce quotient. La *probabilité* et l'idf améliorent la pondération des termes de faible fréquence.

En outre, la mesure *C-value* est fondée sur la fréquence des termes. Le but de *C-value* (Formule 1) est d'améliorer l'extraction de termes imbriqués. Ce critère favorise les termes candidats n'apparaissant pas dans des termes plus longs. Par exemple, dans un corpus spécialisé (ophtalmologie), (Frantzi *et al.*, 2000) ont trouvé le terme non pertinent *soft contact* cependant le terme plus long *soft contact lens* est pertinent.

$$C\text{-}value(A) = \begin{cases} \log_2(|A|) \times f(A) & \text{si } A \notin \text{imbriqué} \\ \log_2(|A|) \times \left(f(A) - \frac{1}{|S_A|} \times \sum_{b \in S_A} f(b) \right) & \text{sinon} \end{cases} \quad (1)$$

Où A est un terme de plusieurs mots, $|A|$ le nombre de mots de A ; $f(A)$ la fréquence du terme A, S_A l'ensemble de termes qui contiennent A et $|S_A|$ le nombre de termes de S_A. Ainsi, *C-value* utilise soit la fréquence du terme si le terme n'est pas inclus dans d'autres termes (première ligne), ou diminue cette fréquence si le terme apparaît dans d'autres termes (deuxième ligne).

Nous avons combiné ces différentes informations statistiques (i.e., *probabilité* des patrons linguistiques, $C\text{-}value$, idf) pour proposer une nouvelle mesure globale de ranking appelée *LIDF-value* (Formule 2). Dans cette formule, A représente un terme de plusieurs mots ; $\mathrm{P}(A_{LP})$ la probabilité du patron linguistique LP associé au terme A qui a la même structure que le patron linguistique LP, c'est-à-dire le poids du patron linguistique LP calculé précédemment.

$$LIDF\text{-}value(A) = \mathrm{P}(A_{LP}) \times idf(A) \times C\text{-}value(A) \quad (2)$$

Ainsi, pour calculer *LIDF-value* nous exécutons trois étapes, résumées ci-dessous :

(1) **Étiquetage grammatical :** nous effectuons l'étiquetage morpho-syntaxique du corpus, ensuite nous considérons le lemme de chaque mot.
(2) **Extraction de termes candidats :** avant d'appliquer les mesures, nous filtrons notre corpus en utilisant les patrons calculés précédemment. Nous choisissons uniquement les termes qui ont une structure syntaxique présente dans la liste de patrons sélectionnés.
(3) **Ranking de termes candidats :** enfin, nous calculons la valeur de *LIDF-value* pour chaque terme.

Afin d'améliorer le classement des termes pertinents, nous proposons, dans la sous-section suivante, de prendre en compte l'information Web.

1. http://www.nlm.nih.gov/research/umls
2. http://nlp.stanford.edu/software/corenlp.shtml

3.2 Une nouvelle mesure de ranking fondée sur le Web : *WAHI*

Des travaux associés à la fouille du Web interrogent les moteurs de recherche pour mesurer l'association entre les mots. Ceci peut être utilisé pour mesurer l'association des mots qui composent un terme (e.g., *soft*, *contact*, et *lens* qui composent le terme pertinent *soft contact lens*). Dans nos travaux, nous proposons d'associer le critère de Dice avec une mesure d'association appelée $WebR$ (Lossio-Ventura *et al.*, 2014) (Formule 3). Par exemple pour le terme *soft contact lens*, le numérateur correspond au nombre de pages Web avec la requête *"soft contact lens"*, et le dénominateur correspond au résultat de la requête *soft ET contact ET lens*.

$$WebR(A) = \frac{nb(\text{“}A\text{”})}{nb(A)} \tag{3}$$

La mesure $WAHI$ (**W**eb **A**ssociation based on **H**its **I**nformation) que nous proposons, combine $Dice$ et $WebR$ de la manière suivante :

$$WAHI(A) = \frac{n \times nb(\text{“}A\text{”})}{\sum_{i=1}^{n} nb(a_i)} \times \frac{nb(\text{“}A\text{”})}{nb(A)} \tag{4}$$

Où a_i est un mot, $a_i \in A$ et $a_i = \{nom, adjectif, mot\ etranger\}$. Nous montrons que les ressources de domaine ouvert, telles que le Web, peuvent être exploitées pour aider l'extraction de termes spécifiques.

4 Expérimentations

4.1 Corpus et Protocole

Dans nos expérimentations, nous utilisons le corpus GENIA [3], qui est composé de 2 000 titres et des résumés d'articles des journaux issus de Medline, avec plus de 400 000 mots. GENIA contient des expressions linguistiques qui font référence à des entités d'intérêt en biologie moléculaire telles que les protéines, les gènes et les cellules. L'annotation des termes techniques couvre l'identification des entités physiques biologiques ainsi que d'autres termes importants. Afin de mettre en place un protocole de validation automatique et de couvrir les termes médicaux, nous créons un dictionnaire qui contient tous les termes d'UMLS ainsi que tous les termes techniques de GENIA. De cette manière nous pouvons évaluer la précision avec un dictionnaire de référence plus complet.

4.2 Résultats

Les résultats sont évalués en termes de *précision* obtenue sur les k premiers termes ($P@k$) pour les deux mesures présentées dans la section précédente.

Résultats de *LIDF-value* : Le tableau 1 compare les résultats de *C-value*, *F-TFIDF-C*, avec notre mesure *LIDF-value*. Le meilleur résultat est obtenu par *LIDF-value* pour l'ensemble des valeurs de k. *LIDF-value* est donc plus performante que les mesures de référence avec un gain en précision de 11 points pour les 100 premiers termes extraits. Ces résultats de précision sont illustrés dans la Figure 2.

	C-value	*F-TFIDF-C*	***LIDF-value***
P@100	0.730	0.770	**0.840**
P@200	0.715	0.725	**0.790**
P@300	0.730	0.723	**0.780**
P@400	0.697	0.705	**0.757**
P@500	0.674	0.694	**0.752**
P@600	0.670	0.687	**0.765**
P@700	0.661	0.686	**0.761**
P@800	0.644	0.669	**0.755**
P@900	0.641	0.649	**0.757**
P@1000	0.635	0.637	**0.746**
P@2000	0.601	0.582	**0.708**
P@5000	0.530	0.513	**0.625**
P@10000	0.459	0.439	**0.574**
P@20000	0.382	0.335	**0.416**

TABLE 1: Précision selon le nombre de termes ($P@k$)

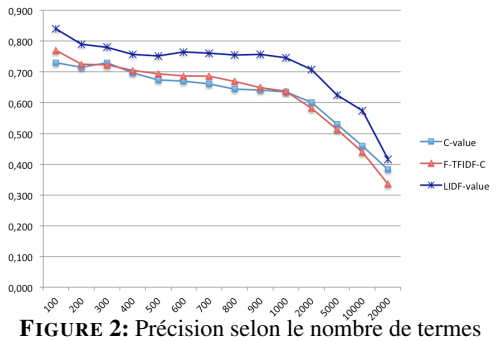


FIGURE 2: Précision selon le nombre de termes

Résultats des indexes de termes : Nous avons évalué *LIDF-value* et les mesures de référence avec une séquence de n-grammes de mots (i.e., un n-grammes de mots est un terme de n mots, par exemple *human immunodeficiency virus* est un 3-grammes de mots). Pour cela, nous construisons un index composé de n-grammes de mots ($n \geq 2$). Nous expérimentons la performance de *LIDF-value* sur les n-grammes de mots en prenant les 1 000 premiers termes. Le Tableau 2 présente la comparaison de la précision pour les 2-grammes, 3-grammes et 4+ grammes de mots.

3. http://www.nactem.ac.uk/genia/genia-corpus/term-corpus

	2-grammes de mots			*3-grammes de mots*			*4+ grammes de mots*		
	C-value	*F-TFIDF-C*	***LIDF-value***	*C-value*	*F-TFIDF-C*	***LIDF-value***	*C-value*	*F-TFIDF-C*	***LIDF-value***
P@100	0.770	0.760	**0.830**	0.670	0.530	**0.820**	0.510	0.370	**0.640**
P@200	0.755	0.755	**0.805**	0.590	0.450	**0.795**	0.455	0.330	**0.520**
P@300	0.710	0.743	**0.790**	0.577	0.430	**0.777**	0.387	0.273	**0.477**
P@400	0.695	0.725	**0.768**	0.560	0.425	**0.755**	0.393	0.270	**0.463**
P@500	0.692	0.736	**0.752**	0.548	0.398	**0.744**	0.378	0.266	**0.418**
P@600	0.683	0.733	**0.763**	0.520	0.378	**0.720**	0.348	0.253	**0.419**
P@700	0.670	0.714	**0.757**	0.499	0.370	**0.706**	0.346	0.249	**0.390**
P@800	0.669	0.703	**0.749**	0.488	0.379	**0.691**	0.323	0.248	**0.395**
P@900	0.654	0.692	**0.749**	0.482	0.399	**0.667**	0.323	0.240	**0.364**
P@1000	0.648	0.684	**0.743**	0.475	0.401	**0.660**	0.312	0.232	**0.354**

TABLE 2: Comparaison de précision des 2-grammes de mots, 3-grammes de mots et 4+ grammes de mots

Résultats de *WAHI* : Notre approche de fouille du Web est appliquée à la fin du processus, avec les 1 000 premiers termes extraits avec les mesures linguistiques et statistiques. La raison principale de cette limitation est le nombre restreint de requêtes autorisées par les moteurs de recherche. À cette étape, l'objectif est de faire le re-ranking des 1 000 termes améliorant la précision par intervalles. Le Tableau 3 montre la précision obtenue après le re-ranking avec *WAHI* et les mesures d'association de référence, utilisant les moteurs de recherche *Yahoo* et *Bing*. Ce tableau souligne que *WAHI* est bien adaptée pour l'EAT obtenant des meilleurs résultats de précision que les mesures de référence.

	YAHOO					*BING*				
	WAHI	*Dice*	*Jaccard*	*Cosine*	*Overlap*	***WAHI***	*Dice*	*Jaccard*	*Cosine*	*Overlap*
P@100	**0.900**	0.720	0.720	0.76	0.730	**0.800**	0.740	0.730	0.680	0.650
P@200	**0.800**	0.775	0.770	0.740	0.765	**0.800**	0.775	0.775	0.735	0.705
P@300	**0.800**	0.783	0.780	0.767	0.753	**0.800**	0.770	0.763	0.740	0.713
P@400	**0.800**	0.770	0.765	0.770	0.740	**0.800**	0.765	0.765	0.752	0.712
P@500	**0.820**	0.764	0.754	0.762	0.738	**0.800**	0.760	0.762	0.758	0.726
P@600	**0.767**	0.748	0.740	0.765	0.748	**0.817**	0.753	0.752	0.753	0.743
P@700	**0.786**	0.747	0.744	0.747	0.757	**0.814**	0.7514	0.751	0.733	0.749
P@800	**0.775**	0.752	0.7463	0.740	0.760	**0.775**	0.745	0.747	0.741	0.754
P@900	**0.756**	0.749	0.747	0.749	0.747	**0.778**	0.747	0.748	0.742	0.748
P@1000	**0.746**	**0.746**	**0.746**	**0.746**	**0.746**	**0.746**	**0.746**	**0.746**	**0.746**	**0.746**

TABLE 3: Comparaison de précision de *WAHI avec YAHOO et BING* et les mesures d'association

Discussion. *LIDF-value* obtient les meilleurs résultats de précision sur tous les intervalles pour l'extraction des termes et pour l'extraction de n-grammes de mots. Le tableau 4 présente les précisions obtenues par nos deux mesures sur le corpus GENIA. *WAHI* basée sur Yahoo obtient une meilleure précision (90 %) pour $P@100$. En comparaison, *WAHI* basée sur Bing obtient une précision de 80 %. Pour les autres intervalles, le Tableau 4 montre que *WAHI* fondée sur Bing obtient en général des résultats légèrement meilleurs. La performance de *WAHI* dépend du moteur de recherche adopté du fait de l'algorithme d'indexation associé. Enfin, le Tableau 4 montre que le re-ranking avec *WAHI* augmente la précision de *LIDF-value*.

	LIDF-value	*WAHI (Bing)*	*WAHI (Yahoo)*
P@100	0.840	0.800	**0.900**
P@200	0.790	**0.800**	**0.800**
P@300	0.780	**0.800**	**0.800**
P@400	0.757	**0.800**	**0.800**
P@500	0.752	0.800	**0.820**
P@600	0.765	**0.817**	0.767
P@700	0.761	**0.814**	0.786
P@800	0.755	**0.775**	**0.775**
P@900	0.757	**0.778**	0.756
P@1000	**0.746**	**0.746**	**0.746**

TABLE 4: Précisions obtenues par *LIDF-value* et *WAHI* selon le nombre de terme ($P@k$)

5 Conclusions et Futurs travaux

L'article présente deux mesures pour l'extraction automatique de termes composés de plusieurs mots. La première est une mesure statistique et linguistique, *LIDF-value*, qui améliore la précision de l'extraction automatique de termes en comparaison avec les mesures classiques. Elle permet de compenser le manque d'information propre à la fréquence avec les valeurs des *probabilités des patrons linguistiques* et *idf*. La seconde, *WAHI*, est une mesure fondée sur le Web et prend comme entrée la liste de termes obtenus avec *LIDF-value*. Cette mesure permet de réduire l'effort humain conséquent

nécessaire à la validation des termes candidats. Nous montrons expérimentalement que *LIDF-value* offre des meilleurs résultats que les mesures de référence pour l'extraction de n-grammes de mots issus du domaine biomédical sur le corpus GENIA. Par ailleurs, ces résultats sont améliorés par l'utilisation de la mesure *WAHI*.

Les perspectives à ce travail sont nombreuses. Tout d'abord, nous souhaitons utiliser le Web pour extraire des termes plus longs que ceux actuellement obtenus. De plus, nous projetons de tester cette approche générale sur d'autres domaines, tels que l'écologie et l'agronomie. Enfin, nous visons à expérimenter notre proposition sur des corpus d'autres langues telles que le français et l'espagnol.

Références

AHMAD K., GILLAM L. & TOSTEVIN L. (1999). University of surrey participation in trec8 : Weirdness indexing for logical document extrapolation and retrieval (wilder). In *TREC*.

BARRON-CEDENO A., SIERRA G., DROUIN P. & ANANIADOU S. (2009). An improved automatic term recognition method for spanish. In *Computational Linguistics and Intelligent Text Processing*, p. 125–136. Springer.

CHAUDHARI D. L., DAMANI O. P. & LAXMAN S. (2011). Lexical co-occurrence, statistical significance, and word association. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, p. 1058–1068, Stroudsburg, PA, USA : Association for Computational Linguistics.

CILIBRASI R. L. & VITANYI P. M. (2007). The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on*, **19**(3), 370–383.

FRANTZI K., ANANIADOU S. & MIMA H. (2000). Automatic recognition of multi-word terms : the c-value/nc-value method. *International Journal on Digital Libraries*, **3**(2), 115–130.

GAIZAUSKAS R., DEMETRIOU G. & HUMPHREYS K. (2000). Term recognition and classification in biological science journal articles. In *Proceeding of the Computional Terminology for Medical and Biological Applications Workshop of the 2 nd International Conference on NLP*, p. 37–44.

JI L., SUM M., LU Q., LI W. & CHEN Y. (2007). Chinese terminology extraction using window-based contextual information. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing07)*, p. 62–74, Berlin, Heidelberg : Springer-Verlag.

KORKONTZELOS I., KLAPAFTIS I. P. & MANANDHAR S. (2008). Reviewing and evaluating automatic term recognition techniques. In *Advances in Natural Language Processing*, p. 248–259. Springer.

KOZAKOV L., PARK Y., FIN T., DRISSI Y., DOGANATA N. & CONFINO T. (2004). Glossary extraction and knowledge in large organisations via semantic web technologies. In *Proceedings of the 6th International Semantic Web Conference and he 2nd Asian Semantic Web Conference (Se-mantic Web Challenge Track)*.

KRAUTHAMMER M. & NENADIC G. (2004). Term identification in the biomedical literature. *Journal of Biomedical Informatics*, **37**(6), 512–526.

LOSSIO-VENTURA J. A., JONQUET C., ROCHE M. & TEISSEIRE M. (2013). Combining c-value and keyword extraction methods for biomedical terms extraction. In *Proceedings of the Fifth International Symposium on Languages in Biology and Medicine (LBM13)*, p. 45–49, Tokyo, Japan.

LOSSIO-VENTURA J. A., JONQUET C., ROCHE M. & TEISSEIRE M. (2014). Biomedical terminology extraction : A new combination of statistical and web mining approaches. In *Proceedings of Journées internationales d'Analyse statistique des Données Textuelles (JADT2014)*, Paris, France.

NEWMAN D., KOILADA N., LAU J. H. & BALDWIN T. (2012). Bayesian text segmentation for index term identification and keyphrase extraction. In *Proceedings of 24th International Conference on Computational Linguistics (COLING)*, p. 2077–2092, Mumbai, India.

PANTEL P., CRESTAN E., BORKOVSKY A., POPESCU A.-M. & VYAS V. (2009). Web-scale distributional similarity and entity set expansion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, p. 938–947, Stroudsburg, PA, USA : Association for Computational Linguistics.

VAN ECK N. J., WALTMAN L., NOYONS E. C. & BUTER R. K. (2010). Automatic term identification for bibliometric mapping. *Scientometrics*, **82**(3), 581–596.

ZADEH R. B. & GOEL A. (2013). Dimension independent similarity computation. *Journal of Machine Learning Research*, **14**(1), 1605–1626.

ZHANG Z., IRIA J., BREWSTER C. & CIRAVEGNA F. (2008). A Comparative Evaluation of Term Recognition Algorithms. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC08)*, Marrakech, Morocco.

Analyse automatique d'espaces thématiques

Gilles Boyé Anna Kupść
Université Bordeaux-Montaigne, Domaine universitaire, 33607 Pessac Cedex
CLLE-ERSS, UMR5263, CNRS, 5 allées Antonio Machado, 31058 Toulouse Cedex 9
gboye@u-bordeaux-montaigne.fr, akupsc@u-bordeaux-montaigne.fr

Résumé. Basé sur les calculs d'entropie conditionnelle de (Bonami & Boyé, à paraître), nous proposons un analyseur automatique de la flexion dans le cadre de la morphologie thématique qui produit le graphe de régularités du paradigme. Le traitement se base sur un lexique de 6440 verbes extraits du BDLex (de Calmès & Pérennou, 1998) associés à leurs fréquences dans Lexique3 (New *et al.*, 2001). L'algorithme se compose de trois éléments : calcul de l'entropie conditionnelle entre paires de formes fléchies, distillation des paradigmes, construction du graphe de régularités. Pour l'entropie, nous utilisons deux modes de calcul différents, l'un se base sur la distribution de l'effectif des verbes entre leurs différentes options, l'autre sur la distribution des lexèmes verbaux en fonction de leurs fréquences pour contrebalancer l'influence des verbes ultra-fréquents sur les calculs.

Abstract. Based on the entropy calculations of (Bonami & Boyé, à paraître), we propose an automatic analysis of inflection couched in the stem spaces framework. Our treatment is based on a lexicon of 6440 verbs present in BDLex (de Calmès & Pérennou, 1998) and associated with their frequencies from Lexique3 (New *et al.*, 2001). The algorithm we propose consists in three steps : computing conditional entropy between all pairs of inflected forms, distilling the paradigms and constructing a regularity graph. For computing entropy, we use two methods : the first one is based on count of verbs in a given distribution whereas the second one takes into account the frequency of each verbal lemma in the distribution to compensate for the bias introduced by the ultra-frequent verbs in the calculation.

Mots-clés : morphologie flexionnelle, espaces thématiques, graphe des régularité, français, verbes.

Keywords: Inflectional morphology, stem spaces, regularity graph, French, verbs.

1 Introduction

Notre travail se situe dans le cadre de la représentation de la flexion en morphologie théorique. Il s'agit ici de présenter un analyseur automatique de la flexion qui produit une représentation des interprédictibilités entre les formes d'un paradigme. Cette représentation dans le cadre de la morphologie thématique prend la forme d'un graphe de régularités (voir Boyé, 2011, pour un exemple de ce type de graphe) qui explicite les liens prédictibles entre les formes.

Notre outil se base sur un lexique de 6440 lexèmes verbaux associés à leurs fréquences. Grâce à BDLex (de Calmès & Pérennou, 1998), une base lexicale de formes fléchies contenant une transcription phonologique de chaque forme, nous constituons le tableau des paradigmes (48 formes par verbe) pour l'ensemble des verbes en API. Les fréquences de lemmes de Lexique3 (New *et al.*, 2001) sont utilisées dans une des deux versions du calcul de l'entropie pour favoriser les généralisations concernant les lexèmes moins fréquents. Ces résultats de prédictibilité basés sur la fréquence donnent une représentation qui pourrait être exploitée pour des études psycholinguistiques dans la lignée de (**?**).

Notre algorithme se compose de trois éléments : le calcul de l'entropie conditionnelle entre paires de formes (une élaboration sur le calcul de Bonami & Boyé, à paraître), la distillation des paradigmes [1], construction du graphe de régularités.

1. Extraction d'un sous-paradigme suffisant pour capter toutes les régularités/irrégularités.

2 Approche thématique

Nous nous situons dans une perspective d'analyse de la flexion au travers des relations entre les formes fléchies. Nous nous basons en partie sur les analyses paradigmatiques développées au sein de l'approche thématique (terme introduit par Plénat, 2008) pour le courant qui considère qu'un lexème est représenté dans le lexique par une collection de radicaux indexés, appelé *espace thématique* (p.ex. Bonami & Boyé, 2003; Boyé & Cabredo Hofherr, 2006; Bonami *et al.*, 2009; Roché, 2010; Tribout, 2012).

Notre analyse morphologique appartient aux approches abstractives de (Blevins, 2006) qui relient directement les formes de surfaces entre elles sans hypothèses sur des formes sous-jacentes, plutôt qu'aux approches qu'il appelle constructives, caractérisées par des formes sous-jacentes et des opérations de construction conduisant aux formes de surface. Le cadre paradigmatique dans lequel nous nous situons se distingue des théories syntagmatiques décrites par (Stump, 2001) avec ses différents modèles d'association entre une forme et ses morphèmes/exposants.

L'approche thématique en morphologie flexionnelle se concentre depuis ses débuts sur les relations entre les formes fléchies au sein des paradigmes mais en se situant au niveau des allomorphies radicales. La question centrale est celle de la correspondance entre les thèmes indexés, autrement dit le remplissage de l'espace thématique.

Nous reprenons ici l'essentiel de cette approche mais en nous situant à la suite de (Bonami & Boyé, à paraître) au niveau des formes fléchies complètes. Nous utilisons, en particulier, l'interprédictibilité des formes fléchies et les notions de lexèmes réguliers et de sous-régularités pour produire un graphe représentant ces informations.

2.1 Interprédictibilité

Dans l'approche thématique, les cases du paradigme dont les contenus sont systématiquement interprédictibles sont formellement associées à un même thème. L'exemple le plus connu en français est celui des formes du futur et du conditionnel qui sont toujours basées sur le même radical quel que soit le verbe considéré. L'ensemble des thèmes nécessaires à la description de la flexion d'une catégorie morphosyntaxique constitue son espace thématique. L'analyse de (Bonami & Boyé, 2003) utilise un espace thématique à 12 cases (thèmes) pour la description de la conjugaison du français. Pour un verbe donné, chaque thème contient un radical qui permet de dériver les formes qui lui sont associées.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}
PASSER	pas	pas	pas	pas	pas	pas	pas	pas	pase	pasə	pasa	pase
BOIRE	byv	bwav	bwa	byv	bwa	byv	bwav	byv	bwa	bwa	by	by
ALLER	al	vɔ̃	va	al	va	al	aj	al	ale	i	ala	ale

TABLE 1 – Exemples de radicaux correspondant aux 12 thèmes.

(Bonami & Boyé, à paraître) défendent une approche abstractive qui abandonne l'hypothèse du thème partagé pour adopter la notion des alliances de formes de (Stump & Finkel, 2013). La notion d'alliance permet d'associer les cases du paradigme dont les contenus sont systématiquement interprédictibles directement entre elles sans supposer de forme sous-jacente commune.

	1sg	2sg	3sg	1pl	2pl	3pl
PRÉSENT	*3*	3	3	*1*	1	*2*
IMPARFAIT	1	1	1	1	1	1
PASSÉ	*11*	11	11	11	11	11
FUTUR	*10*	10	10	10	10	10
SUBJ. PRÉS.	*7*	7	7	*8*	8	7
SUBJ. IMPARF.	11	11	11	11	11	11
CONDITIONNEL	10	10	10	10	10	10
IMPÉRATIF	—	*5*	—	*6*	6	—

INFINITIF	*9*
PART. PRÉSENT	*4*
PART. PASSÉ	*12*

TABLE 2 – Les alliances de formes de la conjugaison du français d'après (Bonami & Boyé, 2003) [2]

2. Les cases qui portent le même numéro font partie de la même alliance.

Par exemple, dans le cas des alliances de la Table 2, l'ensemble des cases {imparfait.1sg, présent.3pl, présent.1sg, part.présent, impératif.2sg, impératif.1pl, subj.prés.1sg, subj.prés.1pl, infinitif, futur.1sg, passé.1sg, part.passé} [3] constituent une distillation du paradigme.

Dans ce cadre, on a l'équivalent de l'espace thématique en conservant une forme représentative pour chaque alliance (parties principales). On obtient ainsi un ensemble réduit de formes, distillation du paradigme (terme introduit par Stump & Finkel, 2013), qui capte l'ensemble des données suffisantes pour prédire systématiquement les paradigmes complets.

2.2 Régularités et sous-régularités

Dans le cas de l'espace thématique comme avec la distillation du paradigme, l'étape suivante consiste à étudier les relations entre les éléments. Dans la suite de cette section, la description repose sur une distillation et ses parties principales. Dans le cadre initial de l'approche thématique, il s'agissait d'observer les relations entre les formes des lexèmes réguliers et de les comparer avec celles des lexèmes dits irréguliers pour identifier les similarités locales (Bonami & Boyé, 2003, est un exemple typique de ce type d'analyse). Par exemple, bien que MORDRE soit irrégulier, le rapport entre *nous mordons* et *ils mordent* est le même que entre *nous bordons* et *ils bordent* pour BORDER. Dans l'approche thématique, l'apparition de ce type de sous-régularités pour un effectif important de lexèmes irréguliers est associée à une notion de proximité entre les formes. Ces rapprochements sont ensuite utilisés pour établir un graphe reliant les éléments de la distillation entre eux.

Autrement dit, dans le cas de la conjugaison du français, pour chaque type de verbes réguliers, les rapports qu'entretiennent ses parties principales entre elles définissent une classe flexionnelle au sens classique [4].

	IPF.1	PRS.6	PRS.1	PCP.PRS	IMP.2	IMP.4	SBJV.1	SBJV.4	INF	FUT.1	PST.1	PCP.PST
Régulier.1	X	X	X	X	X	X	X	X	Xe	Xə	Xa	Xe
Régulier.2	Xis	Xis	Xi	Xis	Xi	Xis	Xis	Xis	Xi	Xi	Xi	Xi
SORTIR	sɔrt	sɔrt	sɔr	sɔrt	sɔr	sɔrt	sɔrt	sɔrt	sɔrti	sɔrti	sɔrti	sɔrti
MORDRE	mɔrd	mɔrd	mɔr	mɔrd	mɔr	mɔrd	mɔrd	mɔrd	mɔrd	mɔrd	mɔrdi	mɔrdy

TABLE 3 – Exemples de sous-régularités.

La comparaison de ces classes flexionnelles régulières avec les paradigmes des verbes irréguliers permet d'identifier les formes correspondant exactement aux mêmes rapports et d'établir toutes les sous-régularités. L'organisation de ces sous-régularités est alors captée sous la forme d'un graphe dont les arcs représentent les sous-régularités. Dans le cas de la Table 3, on observe une sous-régularité qui ne concerne que PRS.1 et IMP.2, on relie donc les deux cases directement par un arc. Pour les sous-régularités concernant des séries plus grandes comme IPF.1, PRS.6, PCP.PRS, IMP.4, SBJV.1, SBJV.4, ci-dessus, on observe les sous-régularités pour l'ensemble des lexèmes et on procède par recoupements pour inférer les arcs. Jusqu'à présent ce type de graphe a été construit manuellement, nous proposons ici une méthode pour le construire automatiquement à partir d'une approche entropique.

3 Approche entropique

(Ackerman *et al.*, 2009) amènent formellement une nouvelle perspective pour la morphologie flexionnelle avec le Paradigm Cell Filling Problem (une version plus générale de la question du remplissage des espaces thématiques) et l'étude des relations entre formes en terme de théorie de l'information (Shannon, 1948). Cette nouvelle approche, basée sur la notion d'entropie conditionnelle, mesure notamment l'incertitude pour le remplissage d'une case du paradigme à partir d'une autre pour l'ensemble des lexèmes. L'entropie est mesurée en bits d'information nécessaires à éliminer l'incertitude.

$$H[X] = -\sum_{x \in X} p(x) \log_2 p(x) \quad (1)$$

Une entropie nulle représente donc l'absence totale d'incertitude, un tirage à pile ou face présente une entropie de 1 et un dé à 6 faces, une entropie de 2,58.

3. La distillation proposée ici est constituée des formes dont les numéros figurent en italique dans la Table 2. Il s'agit de choisir une case représentant chaque alliance, autrement dit une case pour chaque numéro de 1 à 12.

4. Pour des raisons de place, les noms des cases du paradigmes sont abrégés dans la suite : IPF=indicatif imparfait, PRS=indicatif présent, PCP=participe, IMP=impératif présent, SBJV=subjonctif présent, INF=infinitif, PST=passé, 1=1sg ; 2=2sg, 3=3sg, 4=1pl, 5=2pl, 6=3pl.

(Bonami *et al.*, 2011), à la suite de (Ackerman *et al.*, 2009), introduisent l'approche entropique pour comparer la complexité du remplissage des paradigmes entre le français et le mauricien. Leur analyse construite autour du moteur de calcul d'entropie PredSPE développé par Olivier Bonami. Le calcul de l'entropie dans notre travail est basé sur une adaptation de ce moteur.

3.1 Entropie conditionnelle

Pour calculer l'entropie conditionnelle, nous utilisons le mécanisme proposé par (Bonami & Boyé, à paraître). La procédure se décompose en deux étapes. La première est basée sur le Minimal Generalization Learner de (Albright, 2002) appliquée à chaque paire de cases du paradigme :

— pour chaque lexème, on calcule la transformation phonologique qui permet de passer de la forme$_1$ à la forme$_2$ par exemple, pour PRS.6 ⟶ INF :
  ɛn ⟶ ɑ̃dr (PRENDRE, ...)
  Ø ⟶ e (PASSER, ...)

— pour chaque transformation, on calcule la généralisation phonologique minimale du contexte d'application qui couvre l'ensemble des lexèmes concernés d'après la décomposition des phonèmes en traits distinctifs.
  La transformation ɛn ⟶ ɑ̃dr, par exemple, ne s'applique qu'à PRENDRE et ses dérivés par préfixation (APPRENDRE, COMPRENDRE, REPRENDRE, etc.). La généralisation minimale doit couvrir toute forme se terminant prɛn, c'est à dire que le contexte lui-même doit être : *pr—

— la transformation et le contexte donne une règle de correspondance entre la forme$_1$ et la forme$_2$
  ɛn ⟶ ɑ̃dr / *pr—
  Ø ⟶ e / *[p,t,k,b,d,g,f,s,ʃ,v,z,ʒ,m,n,ɲ,j,l,r,w,ɥ,i,y,ɛ,e,ə,u,ɔ,o,ɛ̃,œ̃,ɔ̃]—

La deuxième étape consiste à associer chaque forme$_1$ avec l'ensemble des règles de correspondance dont le contexte d'application est compatible. Ces règles forment un n-uplet caractéristique qui à son tour forme une classe de formes$_1$. On évalue l'entropie locale pour chaque classe, et à partir de ces entropies locales, on obtient l'entropie conditionnelle de la case$_1$ à la case$_2$.

Par exemple, toujours pour PRS.6 ⟶ INF, on a 421 verbes qui sont compatibles avec exactement les deux règles suivantes. Ils forment la classe 7 et leur effectif se répartit comme suit :

417 : Ø ⟶ e / *[p,t,k,b,d,g,f,s,ʃ,v,z,ʒ,m,n,ɲ,j,l,r,w,ɥ,i,y,ɛ,e,ə,u,o,ɔ,ɛ̃,œ̃,ɔ̃]— (ALITER)
4 : Ø ⟶ r / *[m,n,ɲ,j,l,r,w,ɥ,i,y,ɛ,e,ə,a,u,o,ɔ,ɛ̃,œ̃,ɔ̃,ɑ̃][p,t,k,b,d,g,f,s,ʃ,v,z,ʒ,j,r,w,ɥ,i,y,ɛ,e,ə,a,u,o,ɔ]— (INCLURE)

Cette répartition des effectifs (417 vs 4) donne une entropie conditionnelle locale de 0,0775 qui se combine avec celles de toutes les autres classes de PRS.6 à INF pour donner une entropie conditionnelle de 0,288 pour cette paire de cases.

À la fin de cette phase, on obtient un tableau des entropies conditionnelles entre toutes les cases du paradigme. Comme le font remarquer (Bonami & Boyé, à paraître), ce tableau n'est pas symétrique. Par exemple, la case IPF.1 permet de prédire à coup sûr la case IPF.4 tandis que, en sens inverse, l'entropie est très élevée.

3.2 Analyse automatique

Notre analyse automatique, proposée ici, consiste à explorer le tableau des entropies afin de trouver un graphe, qui permette de connecter toutes les cases du paradigme en minimisant les incertitudes, autrement dit, en optimisant l'entropie.

Dans un premier temps, nous procédons à la détection des alliances de formes, c-à-d des formes qui sont interprédictibles (i.e. prédictibles dans les deux sens). En terme d'entropie, cela signifie qu'il n'y a aucune incertitude pour prédire la case$_1$ par rapport à la case$_2$ et inversement. La procédure consiste donc à repérer dans le tableau les paires de cases dont l'entropie conditionnelle est réciproquement nulle. Effectivement, cette méthode nous permet de retrouver, entre autres, l'alliance entre les formes du futur et du conditionnel discutée à la section 2.1.

Ensuite, on procède à la distillation du paradigme. Une fois les alliances identifiées, toutes les formes d'une alliance étant *équivalentes*, on choisit une seule forme comme représentant de chaque alliance. Par exemple, si on garde dans le tableau la forme FUT.1 comme représentant de l'alliance des formes du futur et du conditionnel, toutes les lignes et colonnes qui correspondent aux autres membres de la même alliance, par exemple, FUT.2 ou COND.1, disparaissent.

On obtient ainsi un tableau réduit du même type que le tableau complet mais qui ne contient qu'un représentant pour les formes interprédictibles. On a alors conservé uniquement les formes qui sont indispensables pour (re-)construire le paradigme complet.

Nous cherchons ensuite à identifier des liens entre les cases retenues, afin de construire le graphe des régularités qui nous permettra de connecter toutes les cases du paradigme.

Pour constituer ce graphe, on explore le tableau distillé. Chaque case correspond à un nœud dans le graphe. Le but est de joindre tous les nœuds, en ajoutant les liens (arcs dans le graphe) les plus sûrs. On commence par mettre en place les relations les plus prédictibles c-à-d par identifier les paires de cases dont les entropies conditionnelles sont les plus faibles. Une fois tous les liens d'un niveau (même valeur d'entropie) exploités, on passe au niveau supérieur pour continuer à ajouter des arcs. Un nouvel arc est ajouté si il permet d'atteindre de nouveaux nœuds. Par exemple, si la case_1 et la case_2 sont déjà connectées, ainsi que la case_2 et la case_3, on n'ajoutera pas de nouvel arc pour connecter directement la case_1 à la case_3. La procédure s'arrête quand tous les nœuds peuvent s'inter-atteindre et qu'on a obtenu un graphe orienté connecté.

4 Analyse des résultats

À partir du tableau des entropies conditionnelles obtenu grâce à PredSPE, notre analyse automatique a tracé un premier graphe de régularités (à gauche, Figure 1) basé sur la répartition des effectifs de verbes dans les différentes classes. Parmi les 15 alliances distinguées, 6 reposent sur quelques lexèmes très irréguliers qui jouent un rôle majeur dans le calcul des entropies. Ces lexèmes sont tous très fréquents et leurs formes sont pour la plupart mémorisées. Ils se situent en dehors du système de régularités que nous cherchons à identifier ici. Pour minimiser leur influence, nous avons introduit la fréquence comme un des paramètres du calcul et modifié PredSPE en conséquence. Chaque verbe a été associé à l'inverse de sa fréquence de lemme dans Lexique.org [5]. Les verbes dérivés par préfixation ont été regroupés et associés à la fréquence la plus haute de la famille pour tenir compte de l'effet d'entraînement du modèle sur les autres membres de sa famille. Par exemple, REPRENDRE dont la flexion suit directement le modèle de PRENDRE est associé non pas à sa propre fréquence (126/Mmots) mais à celle de son modèle plus fréquent (1913/Mmots). Ce calcul permet de favoriser les généralisations sur les lexèmes à fréquence réduite et donc augmenter l'importance des régularités recherchées. Le graphe des régularités correspondant à ce calcul est à droite dans la Figure 1.

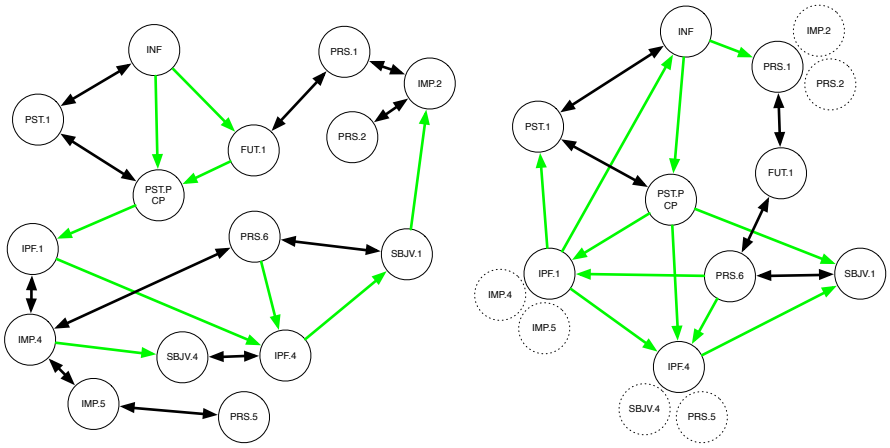


FIGURE 1 – Graphes des régularités basés sur l'entropie avec les effectifs et avec les fréquences [6]

Dans les deux cas, les graphes prennent une forme relativement différente de celles des graphes obtenus manuellement mais on y retrouve des motifs communs. Par exemple, les relations entre INF, PST.PCP et PST.1 ou celle entre FUT.1 et PRS.1 apparaissent aussi bien dans nos deux versions que dans dans le graphe obtenu manuellement (Boyé, 2011). L'introduction des fréquences permet de réduire la distillation de 15 à 9 cases en formant des alliances plus étendues. La réduction de l'influence de verbes extrêmement fréquents, comme ALLER, ÊTRE, AVOIR, SAVOIR, FAIRE et DIRE, a permis de regrouper des alliances du premier graphe dans le second (IPF.1 avec IMP.4 et IMP.5, IPF.4 avec SBJV.4 et PRS.5, PRS.1 avec PRS.2 et IMP.2).

5. Ici, nous avons utilisé la fréquence du lemme dans FranText.

6. Les arcs notés en noir correspondent à des relations d'interprédictibilité symétrique. C'est à dire que la case A et la case B sont les meilleurs prédicteurs l'une de l'autre. Les arcs notés en vert correspondent à des relations asymétriques. C'est à dire que la case A est le meilleur prédicteur de la case B, mais pas l'inverse.

Selon nous, les différences entre les graphes obtenus manuellement et les nôtres tiennent à deux facteurs. D'une part, les sous-régularités recherchées sont différentes. Dans les graphes manuels, on présuppose des modèles réguliers *arbitrairement* et on cherche des analogies similaires pour les irréguliers. Dans notre travail, nous avons cherché à identifier des sous-régularités *sans hypothèse a priori sur des classes régulières*, en étudiant directement toutes les analogies. D'autre part, les relations de nos graphes sont orientées pour tenir compte du fait que la prédictibilité n'est pas symétrique dans le cas général tandis que les relations établies à la main dans le cadre des études thématiques précédentes étaient toutes symétriques par construction, à l'exception de (Bonami & Boyé, 2007) qui, du fait, limitait son étude aux seuls verbes réguliers.

Pour conclure, la méthodologie présentée ici permet donc d'étendre les possibilités de l'approche thématique à des relations plus complexes sur de grands jeux de données. Elle peut s'appliquer à d'autres langues alphabétiques à morphologie affixale. L'outil utilisé peut s'appliquer directement à un lexique flexionnel phonétisable avec, si possible, une base de fréquence pour les lexèmes.

Pour compléter le développement de notre analyseur, il serait nécessaire d'associer automatiquement à chaque arc, les règles de transformation régulières permettant de prédire les correspondances entre formes dans le passage d'une case à une autre. Ceci constituera l'étape suivante de notre travail.

Références

ACKERMAN F., BLEVINS J. P. & MALOUF R. (2009). Parts and wholes : Implicative patterns in inflectional paradigms. In J. P. BLEVINS & J. BLEVINS, Eds., *Analogy in Grammar : Form and Acquisition*, p. 54–82, Oxford : Oxford University Press.

ALBRIGHT A. (2002a). *The identification of bases in morphological paradigms*. PhD thesis, UCLA.

ALBRIGHT A. (2002b). Islands of reliability for regular morphology : evidence from italian. *Language*, **78**, 684–709.

BLEVINS J. P. (2006). Word-based morphology. *Journal of Linguistics*, **42**, 531–573.

BONAMI O. & BOYÉ G. (2003). Supplétion et classes flexionnelles dans la conjugaison du français. *Langages*, **152**, 102–126.

BONAMI O. & BOYÉ G. (2007). Remarques sur les bases de la conjugaison. In E. DELAIS-ROUSSARIE & L. LABRUNE, Eds., *Des sons et des sens : données et modèles en phonologie et en morphologie*, p. 77–90.

BONAMI O. & BOYÉ G. (à paraître). De formes en thèmes. In F. VILLOING & S. DAVID, Eds., *Foisonnements morphologiques. Etudes en hommage à Françoise Kerleroux* : Presses Universitaires de Vincennes.

BONAMI O., BOYÉ G. & HENRI F. (2011). Measuring inflectional complexity : French and mauritian. Présentation au colloque Quantitative Measures in Morphology and Morphological Development, San Diego, 2011.

BONAMI O., BOYÉ G. & KERLEROUX F. (2009). L'allomorphie radicale et la relation flexion-construction. In B. FRADIN, F. KERLEROUX & M. PLÉNAT, Eds., *Aperçus de morphologie du français*, p. 267–286.

BOYÉ G. (2011). Régularités et classes flexionnelles dans la conjugaison du français. In M. ROCHÉ, G. BOYÉ, N. HATHOUT, S. LIGNON & M. PLÉNAT, Eds., *Des unités morphologiques au lexique*, Langues et Syntaxe, p. 41–68 : Hermes Science Publishing.

BOYÉ G. & CABREDO HOFHERR P. (2006). The structure of allomorphy in spanish verbal inflection. *Cuadernos de Lingüística*, **13**, 9–24.

DE CALMÈS M. & PÉRENNOU G. (1998). Bdlex : a lexicon for spoken and written french. In *1st International Conference on Language Resources and Evaluation*, p. 1129–1136. Grenade : ELRA.

NEW B., PALLIER C., FERRAND L. & MATOS R. (2001). Une base de données lexicales du français contemporain sur internet : Lexique. *L'Année Psychologique*, **101**, 447–462.

PLÉNAT M. (2008). Le thème l de l'adjectif et du nom. *Congrès Mondial de Linguistique Française 2008*, p. 139.

ROCHÉ M. (2010). Base, thème, radical. *Recherches linguistiques de Vincennes*, **39**.

SHANNON C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, **27**, 379–423, 623–656.

STUMP G. T. (2001). *Inflectional Morphology*. Cambridge University Press.

STUMP G. T. & FINKEL R. (2013). *Morphological Typology : From Word to Paradigm*, volume 138 of *Cambridge Studies in Linguistics*. Cambridge University Press.

TRIBOUT D. (2012). Verbal stem space and verb to noun conversion in French. *Word Structure*, **5**(1), 109–128.

Extraction et représentation des constructions à verbe support en espagnol

Sandra Milena Castellanos Páez
LIG – GETALP. Grenoble, France.
sandra.castellanos@imag.fr

Résumé. Le traitement informatique de constructions à verbe support (prendre une photo, faire une présentation) est une tâche difficile en TAL. Cela est également vrai en espagnol, où ces constructions sont fréquentes dans les textes, mais ne font pas souvent partie des lexiques exploitables par une machine. Notre objectif est d'extraire des constructions à verbe support à partir d'un très grand corpus de l'espagnol. Nous peaufinons un ensemble de motifs morpho-syntaxiques fondés sur un grand nombre de verbe support possibles. Ensuite, nous filtrons cette liste en utilisant des seuils et des mesures d'association. Bien que tout à fait classique, cette méthode permet l'extraction de nombreuses expressions de bonne qualité. À l'avenir, nous souhaitons étudier les représentations sémantiques de ces constructions dans des lexiques multilingues.

Abstract. The computational treatment of support verb constructions (take a picture, make a presentation) is a challenging task in NLP. This is also true in Spanish, where these constructions are frequent in texts, but not frequently included in machine-readable lexicons. Our goal is to extract support verb constructions from a very large corpus of Spanish. We fine-tune a set of morpho-syntactic patterns based on a large set of possible support verbs. Then, we filter this list using thresholds and association measures. While quite standard, this methodology allows the extraction of many good-quality expressions. As future work, we would like to investigate semantic representations for these constructions in multilingual lexicons.

Mots-clés : Expressions à verbe support, extraction, corpus, expressions polylexicales.
Keywords: Support verb expressions, extraction, corpus, multiword expressions.

1 Introduction

Les locuteurs natifs d'une langue ne se rendent pas compte que l'utilisation d'un certain mot provoque souvent l'utilisation d'un autre, et que ce processus permet de produire une expression correcte et naturelle. Après la lecture de la vaste littérature sur ce phénomène (Firth, 1957; Mel'čuk, 1981; Choueka, 1988; Smadja, 1993), nous adopterons la définition de Manning et Schutze (1999). Une « expression polylexicale » est une expression constituée de deux ou plusieurs mots qui correspondent à une façon conventionnelle de dire les choses et qui est caractérisée par une compositionnalité sémantique limitée, c'est-à-dire que le sens de l'expression ne peut pas être prédit à partir des sens des mots qui la composent.

Nous nous intéressons à une sous-classe spéciale de collocations, les *constructions à verbe support* (CVS)[1]. Les constructions de ce type correspondent à une structure linguistique formée par un verbe et un nom prédicatif. Les verbes *donner*, *faire* et *prendre* font partie de la liste des verbes support inclus dans la grammaire CVS[2] du français. Ils y sont intégrés en raison du peu de contenu sémantique qu'ils apportent à des expressions comme *faire un pas*, *donner une gifle* et *prendre la fuite*. Il y a peu d'éléments dans les sens des verbes *donner*, *faire* ou *prendre* qui nous indiquent la raison pour laquelle nous avons à dire *faire un pas* à la place de **donner un pas*.

Les CVS jouent un rôle important en ce qui concerne les applications concrètes telles que la traduction automatique, la recherche et l'extraction d'informations, les systèmes de questions-réponses, la synthèse, etc. (Laporte et al., 2008). Les CVS ont une fréquence élevée d'apparition dans la langue espagnole (Alvariño, 1999). Par conséquent, le choix du verbe pour un nom est complexe et donc présente des problèmes pour les apprenants de cette langue étrangère. Par exemple, la traduction automatique des CVS vers une autre langue peut donner comme résultat : (1) CVS → CVS ; (2) CVS → Verbe ; (3) Verbe → CVS (Zarco, 1997).

[1] La terminologie de ces verbes est variée : *light verb* (Jespersen, 1965), *funktionsverb* (Von Polenz, 1963), *predicado complejo* (Zarco, 1998).

[2] Cf. Liste des verbes support inclus dans la grammaire CVS (Laporte et al., 2008)

Les principaux objectifs de ce travail sont (1) estimer la présence et l'ubiquité des CVS dans les dictionnaires et les corpus, (2) appliquer des techniques d'extraction de CVS à partir de corpus. Pour des raisons liées à la facilité de l'évaluation des résultats par des locuteurs natifs, nous nous concentrons sur l'espagnol.

2 État de l'art

2.1 Constructions à verbe support

Les CVS sont des expressions lexicalisées, et plus précisément des expressions syntaxiquement flexibles (Sag et al., 2002). Il s'agit d'un syntagme verbal résultant, le plus souvent, d'une combinaison entre un verbe sémantiquement vide et un nom déverbal. Cette structure est soumise à une variabilité syntaxique complète et à un certain degré de compositionnalité sémantique.

1.	(a)	Kim	*gives advice*	to	first year students	2.	(a)	Paul	takes	a walk
	(b)	Kim	*donne un conseil*	aux	étudiants de première année		(b)	Paul	*prend	une promenade
	(c)	Kim	*da un consejo*	a los	estudiantes de primer año		>	*Paul fait une promenade*		
							(c)	Paul	*toma	un paseo
							>	*Paul da un paseo*		

L'exemple 1 décrit la même CVS en anglais, en français et en espagnol. Ici, c'est le verbe *donner* qui se comporte comme verbe support du nom *conseil*. Une traduction mot à mot semble pertinente pour ce cas précis, mais ce phénomène n'est pas toujours tout à fait présent. Dans la CVS de l'exemple 2, le sujet *Paul* peut *faire une promenade* (en français) ou *donner une promenade* (en espagnol) mais il ne pourra jamais la *prendre* comme c'est le cas en anglais.

D'autre part, alors que les CVS acceptent une variabilité syntaxique totale (exemple 3), elles présentent un degré de compositionnalité sémantique qui empêche la formation des CVS alternatives, comme le montre l'exemple 4.

3. John dio una explicación – John a donné une explication

(a)	Una explicación	fue dada	por John (*Passivation)*	
>	Une explication	a été donnée	par John	
(b)	¿Qué	tipo	de explicación	dio John? (*Extraction)*
>	Quelle	type	d'explication	John a donné ?
(c)	John	dio	una reveladora explicación (*Modification interne*)	
>	John	a donné	une explication révélatrice	

4.

(a)	Susan	{ofreció/*dio/*entregó/*regaló} disculpas	a	su padre
(b)	Susan	a {présenté/*montré/*exposé/*proposé} ses excuses	à	son père
>	Susan	s'est excusée	auprès de	son père

Dans ce dernier cas (4), même quand les verbes qui précèdent le nom *disculpas* (*ses excuses*) sont des synonymes dans d'autres contextes, là, seulement le verbe *ofrecer* (*présenter*) correspond à la combinaison correcte, qui permet de donner la signification de « présenter ses excuses à quelqu'un ».

2.2 Présence dans les lexiques

En ce qui concerne les dictionnaires[3], même quand nous pouvons estimer qu'il y a environ 25 000 CVS, il manque de l'information qui permet d'éliminer notamment les ambiguïtés possibles. Pour l'expression *tomar medidas* (*prendre des mesures*), nous pouvons distinguer plusieurs sens possibles, liés aux différents sens du mot mesure (une taille ou une action). Ajouter la fonction lexicale dans laquelle le nom prédicatif est la base de la collocation semble alors une solution intéressante

Cette solution fût d'ailleurs abordée par Mel'čuk (1996, 2003, 2004). Dans le cadre de la Théorie Sens-Texte, il introduit alors le concept de Fonctions Lexicales (FL). Une FL associe une unité lexicale **L** à une autre unité lexicale **L'** laquelle a une relation sémantique envers **L**. En ce qui nous concerne, la FL $Oper_i$ est la plus intéressante car elle est en charge d'associer un nom prédicatif avec un verbe sémantique vide.

Le tableau 1 contient une compilation de quelques CVS trouvés dans les dictionnaires[4]. Les verbes se situent dans la

[3] (DRAE, 2001 ; WordReference, 2008 ; Gran diccionario de la lengua española, 2005)

[4] (DRAE, 2001 ; WordReference, 2008 ; Gran diccionario de la lengua española, 2005)

première colonne. Les significations accompagnées du numéro assigné à chaque entrée (Gran diccionario de la lengua española, 2005) se trouvent dans la deuxième colonne. Et enfin, des exemples sont placés dans la troisième colonne.

Verbe	Signification	Exemples
Hacer	19 Réduire une chose à la signification du nom auquel elle est attachée	Hizo pedazos la carta
	20 Avec quelques noms, exprime l'action des verbes formés à partir de la racine de ces noms.	Le hizo burla
Dar	5 Effectuer l'action indiquée par le nom.	Nos dimos un abrazo
Tener	11 Avec des noms signifient le temps, exprime la durée ou l'âge.	Tiene seis años
Tomar	9 Suivi par certains noms déverbaux, il indique l'action de faire ce que le verbe (d'où les noms dérivent) exprime.	Tomar un baño Tomar una decisión.
	10 Recevoir ou acquérir ce que les noms qui accompagnent signifient.	Tomar fuerza Tomar aliento
Echar	32 Suivi par certains noms, il indique de faire l'action qui est signalé par ceux-ci.	Echarse una siesta.
	34 Suivi par des noms ou des expressions indiquant peine ou sanction, il s'agit de condamner une personne à celles-ci.	Le echaron 5 años de cárcel.

TABLEAU 1 Compilation de quelques CVS trouvés dans les dictionnaires

2.3 Extraction automatique à partir du corpus

Pour ce qui est de l'extraction automatique, divers travaux existent pour identifier les CVS. Il existe des approches qui peuvent soit tenir compte du contexte pour choisir si un candidat est une CVS ou non, soit extraire des paires verbe-objet et après l'utilisation de certaines méthodes indépendantes du contexte, faire son choix. Certaines méthodes sont statistiques, fondées sur la fréquence des mots, tandis que d'autres se fondent sur des règles linguistiques.

Lin (1998) utilise ainsi des techniques statistiques combinées à des traitements syntaxiques. Son travail consiste à collecter des triplets de dépendance, corriger leurs erreurs de comptage et enfin, les filtrer avec leur information mutuelle. Dans la même ligne de conduite, Stevenson et al. (2004), utilisent la même mesure pour détecter les collocations et capturer les propriétés linguistiques de la construction.

Par contraste, Vincze (2013) fait appel à l'utilisation des caractéristiques contextuelles et au modèle des champs aléatoires conditionnels (cf. Lafferty, 2001). Diab et Bhutada (2009) se servent d'un système supervisé pour classifier les combinaisons « verbe + nom » comme des expressions littérales ou idiomatiques, en fonction du contexte.

Finalement, Dias (2003) présente un système hybride qui permet d'extraire des candidats partir d'un corpus étiqueté avec des séquences de partie du discours. Dans son travail, il identifie automatiquement des patrons syntaxiques à partir du corpus, et ensuite, des statistiques de mots sont combinées avec de l'information linguistique pour extraire les séquences de mots les plus intéressantes.

3 Méthodologie

3.1 Constitution du corpus

L'extraction des expressions polylexicales demande le traitement des ressources de grande taille, notre choix a donc ainsi été porté sur l'imposant corpus émanant d'un projet intitulé « GrAF version of Spanish portions of Wikipedia Corpus » (Boleda et Vivaldi, 2012). Cette ressource comporte un corpus en espagnol de 257 019 articles provenant de Wikipédia, qui contiennent environ 150,1 millions de mots au format texte brut. Lors du projet, les auteurs ont nettoyé cette dernière par l'effacement des pages d'homonymie, la suppression de certaines étiquettes XML et l'homogénéisation des étiquettes de terminaison de listes. Ensuite, ils ont ajouté du marquage structurel (tête, paragraphe, phrase, liste, etc.) et de l'information morphosyntaxique.

Cependant, il a été nécessaire de faire du traitement et des transformations de format sur la ressource, en envisageant une utilisation ultérieure de l'outil mwetoolkit (qui sera introduit dans la section ci-dessous) sur le corpus résultant. Dans un premier temps, nous avons créé des scripts pour obtenir des indices de segments à partir du traitement des fichiers fournis. Ensuite, la lemmatisation a été corrigée pour certains segments. Finalement, le format a été transformé et adapté à celui que mwetoolkit reconnaît comme entrée.

À l'issue, nous obtenons un corpus en format XML d'un volume de 4 838 937 segments dont en moyenne 18 mots par segment (d'un total de 88 683 071 mots).

3.2 Extraction des CVS

Nous suivrons la méthodologie décrite dans la Figure 1. Elle est basée sur l'environnement d'extraction d'expressions polylexicales à partir du corpus de l'outil mwetoolkit (Ramisch, 2010).

Tout d'abord, l'outil reçoit comme entrée un corpus prétraité avec le format XML résultant du prétraitement. Ensuite, il emploie une méthodologie qui consiste en une phase d'extraction de candidats, suivie d'une phase de filtrage des candidats. Le système extrait les candidats, en utilisant des patrons morphosyntaxiques spécifiques. Une fois la liste de candidats extraite, il est possible de la filtrer avec des critères simples de seuil, ou des critères plus complexes, tels que leurs mesures d'association. Nous avons traité le corpus dans une phase précédente (sans utiliser l'outil mwetoolkit) pour lui donner le format correct. La mise au point des patrons morphosyntaxiques, employés dans la phase d'extraction des candidats, correspond au résultat de la combinaison entre certains patrons existant dans la littérature (De Miguel, 2008 ; Moncó, 2013) et d'autres extraits à partir de l'analyse d'un pourcentage du texte brut du corpus. La Figure 2 montre un exemple qui permet d'extraire les candidats à être des CVS (ci-après dénommés « candidats CVS ») de la forme « V: hacer »+ « DET : l'Art.Indef. »+ « NC », comme par exemple *hacer una pregunta, hacer una cita, hacer una reserva, hacer una confesión, hacer una llamada, hacer una pausa,* etc.

Dans la phase suivante, nous avons mis en œuvre un filtre heuristique pour garder seulement les candidats qui apparaissent plus de deux fois dans le corpus. La mise en place d'un filtre basé sur les mesures d'association est actuellement en cours.

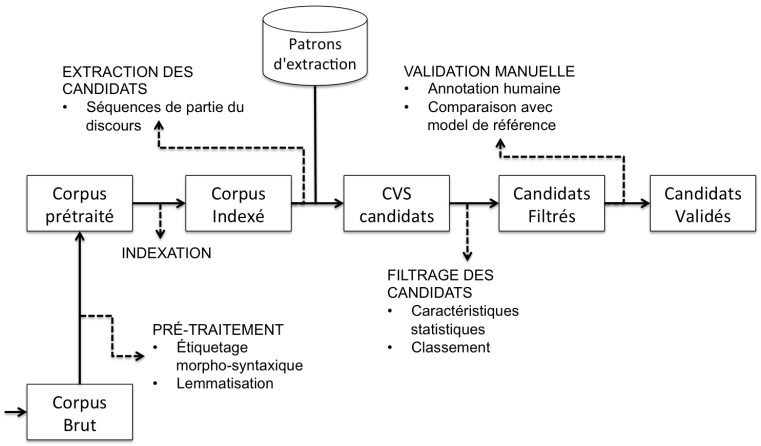


FIGURE 1 Méthodologie d'extraction et de validation des CVS

```
<pat>
        <w pos="V*" lemma="hacer"/>
        <!-- una -->
        <pat repeat="?"><w pos="T.FS*"/></pat>
        <w pos="NC*" />
</pat>
```

FIGURE 2 Patron de la forme V+DET +NC

4 Résultats

La méthodologie et les patrons ont été appliqués au corpus provenant de Wikipédia (Tableau 2). Le tableau 2 présente dans la première colonne la classification des verbes par ordre de pertinence. Les colonnes suivantes présentent la liste des verbes définis comme verbes support et les nombres d'extraction de candidats correspondant à la relation morphosyntaxique prédite par le patron. Ainsi, l'extraction génère un total de 81 274 candidats CVS donc il y a environ 1,7% des segments qui contiennent des candidats CVS. On tire de ce résultat douze verbes dont les plus représentatifs sont *tener* (*avoir*), *hacer* (*faire*) et *dar* (*donner*).

Rang	VS	# cand. à CVS	Rang	VS	# cand. à CVS
12	*Echar*	357	6	*Presentar*	6 220
11	Cometer	510	5	*Tomar*	5 286
10	*Guardar*	1 014	4	*Recibir*	6 135
9	*Sufrir*	2 823	3	*Dar*	11 272
8	*Perder*	2 905	2	*Hacer*	13 673
7	*Ofrecer*	3 603	1	*Tener*	27 476
				Total	**81 274**

TABLEAU 2 Verbes support du corpus

Le tableau 3 présente les 15 meilleurs candidats CVS du corpus, triées par mesure d'association. La première colonne est dédiée à une mesure qui tient compte des candidats composés seulement de 2 mots et les colonnes suivantes tiennent compte des candidats composés de n mots (n > 2). À travers ces trois mesures[5], on peut constater la fréquence élevée des verbes classifiés, nommés ci-dessus. Bien que les deux dernières colonnes contiennent les mêmes candidats, la combinaison avec les candidats extraits pour la mesure `ll` semble importante pour pouvoir amplifier la plage de couverture des candidats CVS. Finalement, on peut remarquer que la nominalisation de quelques candidats est aussi possible, par exemple, *hacer referencia* (faire une référence) peut être remplacé par *referenciar* (se référer), *tomar parte* (prendre part) par *participar* (participer), *dar nombre* (donner un nom) par *nombrar* (nommer), etc. Cette analyse nous permet de voir que l'extraction à partir de corpus est une voie prometteuse pour la constitution d'un lexique électronique de CVS de l'espagnol. La validation manuelle des candidats CVS est actuellement en cours et pourra confirmer cette hypothèse.

`ll`	`t_score`	`mle`
tener lugar	**tener lugar**	**tener lugar**
hacer referencia	**hacer referencia**	**hacer referencia**
dar cuenta	**dar cuenta**	**dar cuenta**
dar lugar	tener una superficie	tener una superficie
hacer cargo	tener un área	tener un área
tomar posesión	tener una población	tener una población
tomar parte	**recibir el nombre**	**recibir el nombre**
hacer prisionero	**dar lugar**	**dar lugar**
tener éxito	**tomar parte**	**tomar parte**
dar origen	**hacer cargo**	**hacer cargo**
dar nombre	tener una longitud	**dar nombre**
dar inicio	**dar nombre**	tener una longitud
tener constancia	**tener éxito**	**tener éxito**
tomar I	**dar origen**	tener forma
dar empereurs	tener forma	**dar origen**

TABLEAU 3 Top-15 candidats CVS triés par leurs mesures d'association (Candidats positifs en gras).

Même si, parmi les premiers candidats, il semble y avoir une proportion d'environ 69% de CVS correctement identifiés, nous voulons vérifier comment cette proportion évolue dans la suite de la liste.

5 Conclusions et travaux futurs

Nous avons montré comment on peut extraire des CVS en espagnol à partir d'un grand corpus. La prochaine étape de ce projet en cours est, dans un premier temps, la validation des candidats. Dans un deuxième temps, nous nous pencherons sur la modélisation de la variabilité des CVS pour représenter ces informations dans le lexique. Puis, dans un troisième temps, nous chercherons à obtenir des représentations sémantiques en utilisant soit des voisins distributionnels, soit des méthodes de paraphrase soit par l'identification du degré de figement ou de variabilité des CVS. Ensuite, dans un quatrième temps, il serait profitable d'évaluer et d'analyser le corpus parallèle pour tenter de trouver des représentations multilingues des CVS.

Références

ALVARIÑO P. (1999). *Sistematización léxico-sintáctica de los predicados complejos*. Tomás Jiménez Juliá, M. Carmen Losada Aldrey, José F. 505-510.

BOLEDA G., VIVALDI J. (2012). *GrAF version of Spanish portions of Wikipedia Corpus.* Universitat Politècnica de Catalunya. Research Group on Natural Language Processing; Universitat Pompeu Fabra. Institut Universitari de Lingüística Aplicada (IULA). http://hdl.handle.net/10230/20047

CHOUEKA, Y. (1988). *Looking for needles in a haystack or locating interesting collocational expressions in large textual databases*. In RIAO'88, 609–624.

DE MIGUEL, E. (2008). Construcciones con verbos de apoyo en español. De cómo entran los nombres en la órbita de los verbos. En Actas del *XXXVII Simposio Internacional de la SEL.* [http://www.unav.es/linguis/simposiosel/actas/]

[5] `ll`: log likelihood; `t_score`: Student's t score; `mle`: Maximum likelihood estimator.

DIAB, M. AND BHUTADA, P. (2009). Verb noun construction MWE token supervised classification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications.* 17-22.

DIAS, G. (2003). Multiword unit hybrid extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18 (MWE '03), Vol. 18.* 41-48.

FIRTH, J. R. (1957). *Papers in Linguistics* 1934-1951. Oxford, UK : Oxford UP. 233.

GRAN DICCIONARIO DE LA LENGUA ESPAÑOLA (2005). Barcelona, España: Larousse.

JESPERSEN, O. (1965). *A Modern English Grammar on Historical Principles*, Part VI, Morphology. London: George Allen and Unwin Ltd.

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.* 282- 289.

LAPORTE E., RANCHHOD E., AND YANNACOPOULOU A. (2008). *Syntactic variation of support verb constructions*. Lingvisticae Investigationes, 31(2):173–185. DOI: 10.1075/li.31.2.04lap.

LIN, D. (1998). Extracting collocations from text to corpora. In *Proceedings of the First Workshop on Computational Terminology.* 57-63.

MANNING C. AND SHUTZE H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, USA : MIT Press. 620p.

MEL'ČUK, I. A. (1981). *Meaning-text models: a recent trend in Soviet linguistics*. The Annual Review of Anthropology.

MEL'ČUK, I. A. (1996). Lexical functions : A tool for the Description of Lexical Relations in the Lexicon. In : Wanner. Leo (ed.), *Lexical Functions in Lexicography and Natural Language Processing*, Amsterdam/Philadelphia : Benjamins, 37-102.

MEL'ČUK, I. A. (2003). Collocations dans le dictionnaire. In : Thomas Szende (réd.), *Les écarts culturels dans les Dictionnaries bilingues*, Paris : Honoré Champion, 19-64.

MEL'ČUK, I. A. (2004). *Verbes suppports sans peine*. Lingvisticae Investigationes, 27 :2, 203-217.

MONCO S. (2013). Adquisición de las construcciones con el verbo «hacer», enfoque plurilingüe. *Revista Nebrija de Lingüística Aplicada* 13 (numéro spécial).

RAMISCH C., VILLAVICENCIO A., BOITET C. (2010). mwetoolkit: a Framework for Multiword Expression Identification. *Proceedings of the Seventh International Conference on Language Resources and Evaluation*. Valetta, Malta.

REAL ACADEMIA ESPAÑOLA. (2001). Diccionario de la lengua española (22.a ed.). [http://www.rae.es/rae.html]

SAG, I. A., BALDWIN, T., BOND, F., COPESTAKE, A., & FLICKINGER, D. (2002). Multiword expressions: A pain in the neck for NLP. *Proceedings of Computational Linguistics and Intelligent Text Processing: Third International Conference: CICLing-2002*, Lecture Notes in Computer Science, 2276, 1-15.

SMADJA, F. A. (1993). *Retrieving collocations from text: Xtract.* Comp. Ling., 19(1):143– 177.

STEVENSON, S., FAZLY, A., AND NORTH, R. (2004). Statistical Measures of the Semi-Productivity of Light Verb Constructions. In *Second ACL Workshop on Multiword Expressions : Integrating Processing*. 1-8.

VINCZE V., NAGY I., ZSIBRITA J. (2013). Learning to detect english and hungarian light verb constructions. *ACM Trans. Speech Lang. Process.* 10, 2, Article 6 (June 2013), 25 pages.

VON POLENZ, P. (1963). *Funktionsverben im heutigen Deutsch.* Düsseldorf : Wirken-des Wort, Beiheft 5.

WORDREFERENCE.COM ONLINE LANGUAGE DICTIONARIES. (2008). Avalaible on : http://www.wordreference.com/

ZARCO TEJADA, Mª A. (1997). Codificación en el lexicón de las relaciones de concurrencia. *Philologia Hispalensis* 11, 83-93.

ZARCO TEJADA, Mª A. (1998). *Predicados complejos y Traducción automática.* Cádiz : Servicio de Publicaciones de la Universidad de Cádiz, 285.

Sous-catégorisation en *pour* et syntaxe lexicale

Benoît Sagot[1] Laurence Danlos[1] Margot Colinet[1,2]
(1) Alpage, INRIA & Université Paris-Diderot, 75013 Paris
(2) LLF, CNRS & Université Paris-Diderot, 75013 Paris
{prenom.nom}@inria.fr

Résumé. La sous-catégorisation d'arguments introduits par la préposition *pour* a été sous-étudiée par le passé, comme en témoigne l'incomplétude des ressources lexico-syntaxiques existantes sur ce point. Dans cet article, nous présentons rapidement les différents types de sous-catégorisation en *pour*, qui contrastent avec les emplois de *pour* comme connecteur de discours. Nous décrivons l'intégration des arguments en *pour* au lexique syntaxique Lefff, enrichissant ainsi les informations de sous-catégorisation de nombreuses entrées verbales, nominales, adjectivales et adverbiales.

Abstract. Sub-categorized arguments introduced by the French preposition *pour* has been under-studied in previous work, as can be seen from the incompleteness of existing lexical-syntactic resources in that regard. In this paper, we briefly introduce the various types of sub-categorization in *pour*, which are to be distinguished from occurrences of *pour* as a discourse connective. We describe how we added arguments in *pour* within the syntactic lexicon Lefff, thus refining sub-categorization information for many verbal, nominal, adjectival and adverbial entries.

Mots-clés : Sous-catégorisation, Arguments en *pour*, Lexiques syntaxiques.

Keywords: Sub-categorization, Arguments in *pour*, Syntactic lexicons.

1 Introduction

Le développement de ressources lexicales syntaxiques pour le français est encore aujourd'hui un champ de recherche ouvert, comme en attestent les nombreux efforts réalisés dans cette direction par diverses équipes, parfois depuis plusieurs dizaines d'années. De telles ressources constituent des bases de données linguistiques descriptives, mais sont également utiles pour différentes tâches de TAL, et notamment en analyse syntaxique symbolique (Riezler *et al.*, 2002; Villemonte De La Clergerie, 2013) et même statistique (Collins, 1997; Mirroshandel *et al.*, 2013). Elles sont enfin au cœur de nombreux travaux à l'interface entre la syntaxe et d'autres niveaux d'analyse linguistique, et notamment la sémantique et le discours.

Nous nous intéressons dans ce travail aux syntagmes prépositionnels en *pour*, en mettant l'accent sur les infinitives. Ces syntagmes sont intéressants pour plusieurs raisons. Tout d'abord, et bien que dans certains cas au moins ils constituent clairement des arguments d'éléments prédicatifs (verbes, noms, adjectifs, voire adverbes), ils ont été relativement peu étudiés et sont assez mal représentés au sein des différentes ressources actuelles : tables du lexique-grammaire (Gross, 1975; Tolone, 2011), Dicovalence (van den Eynde & Mertens, 2006), le Lexique des Verbes Français (Dubois & Dubois-Charlier, 1997) ou le Lefff (Sagot, 2010). Ceci vient au moins en partie de ce que les tests les plus élémentaires pour distinguer arguments et modifieurs reposent sur la cliticisation, opération qui n'est jamais possible directement à partir d'un argument réalisé sous la forme d'un syntagme prépositionnel en *pour*.

Par ailleurs, l'étude des compléments en *pour*, et singulièrement des infinitives en *pour*, est indispensable à l'interface entre syntaxe et discours. Ainsi, dans le cadre du développement du French Discourse TreeBank (FDTB) (Danlos *et al.*, 2012), la première étape consiste à repérer toutes les occurrences de connecteurs du discours. Ceci suppose, pour les unités lexicales qui peuvent avoir un rôle de connecteur, de savoir faire le départ entre les occurences où elles sont connecteurs et les occurrences où elles ne le sont pas. Colinet *et al.* (2014) ont étudié les emplois de *pour* comme connecteur de discours lorsqu'il introduit une infinitive, en mettant au point un ensemble de critères formels permettant de les distinguer des emplois comme introducteur d'infinitives sous-catégorisées. Outre les emplois comme connecteurs (1a) et les emplois sous-catégorisés (1b), cette étude, sur laquelle repose l'amélioration du Lefff présentée ici, a identifié deux autres types d'emplois plus spécifiques : ceux introduisant une « relative sans mot QU » (Huddleston & Pullum, 2002) (1c) et les emplois métadiscursifs, dont trois exemples sont donnés en (1d).

(1) a. Luc a fait une bouillabaisse pour faire plaisir à Marie.
b. Il a suffi d'un café à Marie pour se remettre d'aplomb.
c. Un pont pour franchir l'Amazone a été construit en 1745.
d. Pour conclure. Pour ne citer que lui. Pour le dire autrement.

L'objectif du travail présenté ici est double : (i) réaliser une étude comparative du traitement des différents types d'arguments en *pour* dans le lexique-grammaire, Dicovalence, le Lexique des Verbes Français et le Lefff ; et (ii) intégrer le résultat dans le Lefff, en s'aidant notamment des résultats de l'annotation manuelle des occurrences de *pour* (Colinet *et al.*, 2014) dans le French TreeBank (Abeillé *et al.*, 2003).

Cet article est organisé comme suit. Dans un premier temps, nous passons en revue rapidement certains travaux antérieurs sur la sous-catégorisation en *pour* (section 2). Ensuite, nous présentons de façon succincte mais systématique les différents types de sous-catégorisation en *pour*, et nous les différencions des emplois de *pour* comme connecteur de discours (section 3). Nous présentons enfin la façon dont nous avons intégré ces résultats au lexique Lefff (section 4).

2 Travaux antérieurs

Outre les travaux de Cadiot (1990a; 1990b) et la grammaire de Wyler (2014), les ressources que nous avons étudiées sont les tables du lexique-grammaire (Gross, 1975; Tolone, 2011), Dicovalence (van den Eynde & Mertens, 2006), le Lexique des Verbes Français (Dubois & Dubois-Charlier, 1997) et le Lefff (Sagot, 2010). Mais dans toutes ces ressources, la sous-catégorisation (verbale, nominale, adjectivale, adverbiale) en *pour* y est mal décrite, notamment dans les nombreux cas où *pour* alterne avec *à* ou avec *de*, où souvent seul *à* ou *de* est mentionné. Il en est ainsi en (2) et (3), peut-être sous l'influence du procédé de cliticisation disponible, qui est le procédé habituel pour les objets indirects en *à* ou *de*.

(2) Luc a préposé Max à/pour garder l'entrepôt. / Luc l'y a préposé.
(3) Luc complimente Max de/pour avoir fini à temps. / Luc l'en a chaleureusement complimenté.

Dans le Lexique des Verbes Français (LVF) (Dubois & Dubois-Charlier, 1997), la préposition *pour* est indiquée dans les codes de sous-catégorisation par deux lettres distinctes : *k* lorsqu'elle alterne avec *contre*, et *q* sinon. Seuls 34 verbes sont répertoriés comme sous-catégorisant un argument en *pour/contre*. Tous font partie de la classe C1 des verbes de communication à sujet humain ou animal, avec l'opérateur « loq PR/CT » signifiant « parler pour (PR) ou contre (CR) ». De manière générale, on peut du reste noter que si *pour* permet (généralement) d'introduire un argument nominal, infinitif ou phrastique, *contre* n'introduit que des arguments nominaux. On peut également remarquer que, dans ces cas d'alternance *pour/contre*, ces prépositions peuvent généralement être employées sans objet (*J'ai voté pour*). À côté de ces 34 verbes, 280 autres entrées verbales sous-catégorisant un argument en *pour* sont répertoriés, où *pour* n'alterne pas avec *contre*.

Dans les Tables verbales du Lexique-Grammaire, des arguments en *pour* sont indiqués de deux façons :
– ils peuvent faire l'objet d'une propriété dédiée dont la validité est associée à chaque entrée d'une des classes. C'est par exemple le cas de la propriété N_0 *V* N_1 *à* N_2 *pour Dnum Nmonnaie* codée dans la table 36DT [1], qui est associée par exemple aux verbes *acheter*, *brader*, etc. (cf. *Pierre a bradé ce livre (à Marie) pour 2 euros*). On retrouve également les propriété $Prép_2$ = *pour* et propriété $Prép_3$ = *pour* dans la table 38RR [2]).
– ils peuvent être directement indiqués dans les tables, sous la forme d'un codage explicite de la préposition *pour*. Par exemple, dans la table 1 (verbes à un argument post-verbal prépositionnel qui peut être phrastique), la préposition introduisant l'argument post-verbal est indiquée explicitement. Ainsi, dans cette table, l'entrée du verbe *insister* indique que cette préposition est *pour* (cf. *Pierre a insisté pour partir / pour que Marie parte*).
Au total, 176 entrées verbales dans les tables font usage d'une façon ou d'une autre de la préposition *pour*.

Dans Dicovalence, les arguments en *pour* sont indiqués dans le « paradigme prépositionnel ». Seules 103 entrées incluent un argument en *pour*, dont 9 en alternance avec *contre*. Parmi les 94 autres, 37 sont indiqués comme pouvant être réalisés par (la pronominalisation d')une infinitive.

Enfin, dans le Lefff (version 3.3), 116 entrées verbales sous-catégorisent un argument dont une réalisation en *pour* est possible. Il s'agit pour la plupart d'entrées obtenues par fusion avec Dicovalence ou le LVF (Sagot & Danlos, 2012). On trouve également 46 adjectifs, 5 noms et 4 adverbes qui sous-catégorisent un argument permettant une réalisation en *pour*.

La sous-catégorisation non-verbale d'arguments en *pour*, c'est-à-dire les arguments en *pour* de prédicats nominaux, adjectivaux ou adverbiaux a été quant à elle encore moins étudiée.

1. Classe des verbes « datifs » (structure de base N_0 *V* N_1 *à* N_2, avec *à* N_2 cliticisable en *lui* ; cf. *Zoé donne un livre à Max / Zoé lui donne un livre*).
2. Classe des verbes « résiduels doubles » (structure de base N_0 *V* N_1 *Prép* N_2 *Prép* N_3 et qui ne rentrent pas dans une autre classe plus spécifique).

3 La sous-catégorisation en *pour*

3.1 Critères et annotation des données du French TreeBank

Comme indiqué précédemment, l'un des principaux obstacles à l'étude de la sous-catégorisation en *pour* est la difficulté qu'il y a à les identifier formellement. En effet, si les « relatives sans mot QU » et les *pour* introducteurs d'expressions métadiscursives sont faciles à identifier, la distinction entre *pour* connecteurs de discours et *pour* introduisant un argument sous-catégorisé n'est pas aisée. Il s'agit en effet d'une instance particulièrement délicate du problème général de la distinction entre arguments et modifieurs.

Colinet *et al.* (2014) rappellent deux critères classiques pour distinguer les arguments des modifieurs (critères 1 et 2 ci-dessous), et en proposent d'autres qui s'appliquent spécifiquement au cas de la préposition *pour*. Certains de ces critères s'appliquent correctement à toutes les catégories prédicatives, d'autres ne concernent que la sous-catégorisation verbale.

1. Un argument est *sémantiquement obligatoire*, pas un ajout (on notera que le caractère sémantiquement obligatoire n'implique pas que sa réalisation en syntaxe soit obligatoire). On peut tester ce critère en tentant de supprimer l'argument en *pour* et en vérifiant l'acceptabilité et le changement éventuel de sens. Ainsi, dans *Luc a fait une bouillabaisse pour faire plaisir à Marie*, l'infinitive est effaçable sans changement d'acceptabilité. En revanche, dans *Luc s'est dépêché pour finir*, l'effacement de l'infinitive conduit à un énoncé dont l'interprétation nécessite de récupérer l'argument effacé dans le contexte discursif gauche.
2. Un ajout est plus facilement antéposable qu'un argument. Il s'agit d'une tendance et non d'un critère strict.
3. Commutation avec *à*, *de*, *contre*. Comme évoqué précédemment, *pour* commute dans certains cas avec l'une de ces prépositions introduisant un complément considéré par les ressources existantes comme sous-catégorisé. Nous y revenons plus en détail ci-dessous.
4. En général, lorsque le connecteur de discours *pour* introduit une infinitive, cette infinitive est contrôlée par le sujet (le sujet de l'infinitive est coréférent au sujet de la principale). Si donc on est face à un cas de contrôle d'une infinitive en *pour* par un argument autre que le sujet, il y a de fortes chances qu'il s'agisse d'un cas de sous-catégorisation.
5. De manière générale, on ne peut rajouter la séquence « *et ce,* » avant *pour* que dans les cas où *pour* est connecteur (cf. *Luc a fait une bouillabaisse et ce, pour faire plaisir à Marie* vs. **Luc s'est dépêché et ce, pour finir*). Certains cas sont toutefois moins clairs (*?Luc a pris un taxi et ce, pour aller à l'aéroport*).
6. Les possibilités de réfutation diffèrent selon que *pour* introduit un argument ou un modifieur. Ainsi, si l'on répond « *c'est faux* » à une assertion comme *Luc s'est dépêché pour finir*, on affirme que Luc ne s'est pas dépêché. À rebours, si l'on répond « *c'est faux* » à une assertion telle que *Luc a fait une bouillabaisse pour faire plaisir à Marie*, on affirme soit que Luc n'a pas fait de bouillabaisse, soit qu'il en a faite une mais dans un autre but que celui de faire plaisir à Marie. Nous ne rentrerons pas dans les détails, mais d'autres questions sémantiques liées aux interrogatives et aux propriétés de factualité vont dans le même sens.
7. Si l'on dispose d'un exemple où deux arguments en *pour* sont rattachés au même verbe sans être coordonnés, alors le premier est susceptible d'être sous-catégorisé, et le second est nécessairement modifieur (autrement dit, le second *pour* est un connecteur de discours).

Grâce à ces critères, Colinet *et al.* (2014) ont annoté manuellement toutes les occurrences de *pour* introduisant une infinitive dans le French TreeBank. Les syntagmes prépositionnels nominaux ont été laissés de côté, l'annotation posant un problème supplémentaire, à savoir le fait de déterminer si le GN introduit par pour référé à une éventualité ou non. Sur les 1161 occurrences ainsi annotées, 558 introduisent des compléments sous-catégorisés, 518 sont des connecteurs de discours, 52 introduisent des « relatives sans mot QU », 33 introduisent des expressions métadiscursives.

3.2 Sous-catégorisation verbale en *pour*

Nous avons confronté les résultats de cette annotation aux données fournies par les différentes ressources lexicales mentionnées ci-dessus, en ayant parfois eu recours de façon complémentaire à l'introspection et à des recherches sur internet. Nous en sommes arrivés à l'inventaire suivant d'arguments sous-catégorisés en *pour*, qui reprend le découpage des emplois verbaux selon les tables du lexique-grammaire. [3] Dans la suite, la notation V^i*-inf* dénote une infinitive dont le sujet est coréférent au i-ième argument du verbe (0 pour le sujet, par exemple), et *W* dénote toute séquence de dépendants (par exemple, V^0*-inf W* dénote une infinitive complète dont le sujet coréfère au sujet du verbe de la principale).

3. Faute de place, nous ne discutons pas ici des locutions verbales figées sous-catégorisant un argument en *pour*.

Table 1 : N_0 V pour V^0-inf W La majorité des cas d'alternance *pour/contre* rentrent dans ce cadre là (*batailler*, *se battre*, *manifester*...). On peut également citer un exemple d'alternance *à/pour* (*œuvrer*) et un exemple d'alternance *de/pour* (*se dépêcher*). On trouve également dans le FTB des cas sans alternance (*foncer*, *se précipiter*, *s'endetter*, *intervenir*...), y compris des verbes symétriques dans leur emploi à sujet conjoint (N_0*plur se retrouvent pour* V^0*-inf W* / N_0 *se retrouve avec* N_1 *pour* V^{0+1}*-inf W*), une entrée lexicale excluant l'assertif positif (*ne pas se gêner*) et quelques cas un peu particuliers (*aller (pour)* et *être (pour)* au sens de *être sur le point (de)*, *être parti (pour)*).

Table 2 : N_0 V Loc N_1 pour V^0-inf W Seuls trois exemples trouvés dans le FTB : *venir*, *se rendre* et *sortir* (ce dernier imposant que la préposition *Loc* soit *de*). L'infinitive peut être réalisée avec ou sans la préposition *pour*.

Table 3 : N_0 V N_1 Loc N_2 pour V^1-inf W Seul un exemple trouvé : *envoyer*. Ici aussi, l'infinitive peut être réalisée avec ou sans la préposition *pour*.

Table 9 : N_0 V à N_1 (qu P / pour V^1-inf W) Quelques verbes de communication (*écrire*, *s'adresser*...) et quelques cas particuliers (*recourir*, *s'en remettre*, *se substituer*).

Table 11 : N_0 V (à ce qu P / à/pour V^i-inf W) Les deux types de contrôle sont attestés dès lors que l'argument en *pour* est une infinitive : $i = 0$ (contrôle par le sujet)) et $i = 1$ (contrôle par l'objet). Exemples de contrôle par le sujet : *employer*, *nommer*, *utiliser* ; exemples de contrôle par l'objet : *utiliser* (ambigu avec un contrôle par le sujet). Dans certains cas, seule la préposition *pour* est possible, et *à* est impossible : il en est ainsi par exemple de *choisir*, *désigner*, *retenir* (contrôle par le sujet) ou de *choisir* (qui est donc ambigu), *délaisser*, *invoquer* (contrôle par l'objet). Un cas spécifique est celui de verbes dont l'objet direct non humain est introduit par un déterminant possessif référant obligatoirement au sujet (*Max a dévoué sa/*ma vie à/pour soigner les bêtes*).

Tables 12 et 13 : N_0 V N_1 de/pour V^1-inf W Seuls des verbes sémantiquement proches de *complimenter*, *punir*, *récompenser* sont attestés.

Table 16 : N_0 V Prép N_1 pour V^i-inf W Les deux types de contrôle sont attestés dès lors que l'argument en *pour* est une infinitive : $i = 0$ (contrôle par le sujet)) et $i = 1$ (contrôle par l'objet). Plusieurs cas selon la valeur de *Prép* : avec *Prép = sur* on trouve par exemple *s'appuyer*, *compter* ; avec *Prép = de* on a notamment *profiter* ; avec *Prép = auprès de* on peut citer *s'activer* ; avec *Prép = avec* on retrouve notamment les verbes symétriques dont la construction à sujet conjoint a été mentionnée au titre de la table 1.

Table 18 : N_0 V Prép N_1 Prép N_2 (de/pour V^1-inf W / que Psubj) Plusieurs exemples hétérogènes, comme *enrôler (*N_1 *Loc* N_2 *pour* V^1*-inf W)* ou *prétexter (de* N_1 *auprès de* N_2 *pour* V^1*-inf W)*.

Falloir* et *suffire On trouve un certain nombre d'attestations relevant des types suivants :
- *falloir* N_1*durée (à* N_1*) pour* V^1*-inf W* (*Il a fallu deux heures à Luc pour finir* ; cf. également le point suivant),
- *falloir (V-inf W / qu P /* N_1*) pour V-inf W* (*Il faut manger pour vivre*),
- *suffire (à* N_1*) (de* V^1*-inf W / qu P / de* N_2*) pour* V^1*-inf W* (*Cela suffit à Marie pour être heureuse*),

***Pour* attributifs** *passer pour*, *prendre* N_1 *pour*, etc.

Cas avec restrictions fortes sur le N_1 Quelques cas sont fortement liés à des classes de noms en position d'objet :
- N_0argent/attribut : *avoir*, *débourser*, *dépenser*, *emprunter*, et d'autres ; parallèlement, on trouve *donner* (avec un *à* N_2 et un contrôle par le N_2), *obtenir* et *recevoir* (avec un *de* N_2 et un contrôle par le N_2) et *manquer* (avec un *à* N_1 et pas de contrôle prédéfini).
- N_0durée/date : *mettre*, *passer*, *se donner*, *prendre* une durée pour faire quelque chose, *donner* une durée à quelqu'un pour faire quelque chose, *disposer* d'une durée, *avoir jusqu'à*, *attendre (jusqu'à)* une date, *attendre* une durée pour (avant de) faire quelque chose.

3.3 Sous-catégorisation nominale, adjectivale et adverbiale en *pour*

La plupart des **noms** qui sous-catégorisent un argument en *pour* peuvent se regrouper en quelques classes sémantiques :
- Noms de type *méthode*, comme *moyen*, *mesure*, *solution*. L'argument en *pour* (nominal, infinitif ou phrastique) alterne avec un argument nominal en *de* ; pour *moyen*, il peut être infinitif ou phrastique (en *que* et non *de ce que*).
- Noms de type *combat*, comme *lutte*, *bataille*, *négociations*, *pression*, *surenchère*. Ici *pour* alterne le plus souvent avec *contre* (cf. cependant *concurrence*). Dans certains cas, on peut identifier un autre argument en *avec*.
- Noms de type *qualité*, comme *atout(s)*, *imagination*.
- Noms de type *effort*, comme *intervention*, *tentative*, *démarche(s)*
- Noms de type *réunion*, comme *rassemblement*, *commission*, *partenariat*, *accord*. Dans certains cas *pour* alterne avec *contre*. Dans certains cas également, on peut identifier un autre argument en *de* (et parfois *entre*) voire un troisième argument en *avec* (cf. *l'accord des parties / entre les parties / de X avec Y pour V-inf W*).

– Noms de type *aide*, comme *assistance*, *contribution*. La préposition *pour* alterne souvent avec *à*. On peut identifier un autre argument en *de* (cf. *l'aide de Marie à/pour la bonne marche de la cérémonie*).
– Noms de type argent ou temps (cf. ci-dessus) : *dépenses*, *dette*, *échéance*. Autres compléments sont identifiables.
– Autres classes : noms de type *candidat*, *volontaire* ; noms de type *mandat*, *mission* (avec autre argument en *de*) ; noms de type *appel*, *ultimatum* (avec autre argument en *de*). Dans certains cas, *pour* alterne avec *à*.
– Autres cas : *argument*, *problème*, *condition*…

Un certain nombre d'**adjectifs** ont également été identifiés, dont les plus fréquents sont *nécessaire*, *indispensable*, *idéal*, *suffisant* et *insuffisant*. Enfin, quelques **adverbes** sous-catégorisent également un argument en *pour* : *suffisamment*, *insuffisamment*, *trop* et *assez*.

4 Enrichissement du Lefff

4.1 Le Lefff

Le Lefff (Lexique des Formes Fléchies du Français) (Sagot, 2010) est un lexique morphologique et syntaxique librement disponible. Il est développé dans le cadre du formalisme Alexina, en partie au moyen de techniques automatiques ou semi-automatiques, mais avec une forte composante manuelle. Le Lefff est conçu pour être linguistiquement pertinent tout en pouvant être directement utilisé dans des systèmes de TAL. Il repose sur la notion de redistribution : le lexique *intensionnel*, édité par les développeurs du lexique, associe à chaque entrée un cadre de sous-catégorisation initial et liste les redistributions possibles à partir de ce cadre. Le processus de *compilation* du Lefff intensionnel en Lefff *extensionnel* construit différentes entrées pour ces différentes redistributions et pour chacune des formes fléchies compatibles.

Considérons par exemple l'entrée intensionnelle simplifiée suivante, tirée de la version 3.3 du Lefff :

délaisser$_1$ *Lemma;v;<Suj:cln|sn,Obj:(cla|seréc|sn|seréfl),Obl:(pour-sn|pour-sinf|pour-scompl)>;@CtrlSujObl;*
%actif,%passif,%se_moyen,%ppp_employé_comme_adj
 # Lien externe: Dicovalence(délaisser$_{25480}$*) # Exemple: Paul a délaissé sa famille pour aller se battre*

Cette entrée décrit un verbe ayant trois arguments : un argument dont la fonction syntaxique est la fonction sujet, qui peut être réalisé par un clitique ou un syntagme nominal ; un objet direct dont la réalisation est facultative (liste entre parenthèses) et qui peut prendre la forme d'un clitique, d'un syntagme nominal ou d'un *se* (réfléchi ou réciproque) ; et enfin un argument oblique en pour, nominal ou phrastique (infinitive ou complétive). L'information selon laquelle l'argument oblique, s'il est infinitif, est contrôlé par le sujet est donnée également. Enfin, la liste des redistributions admissibles (actif, passif, *se* moyen, participe passé employé comme adjectif) est indiquée. Une équivalence avec l'entrée correspondante de Dicovalence est donnée, ainsi qu'un exemple (tiré, pour cette entrée, ici de Dicovalence).

4.2 Intégration des arguments en *pour*

À partir de l'étude évoquée à la section précédente, nous avons complété manuellement les entrées lexicales du Lefff avec les arguments réalisables en *pour*, ou avons rajouté à des arguments existants les réalisations en *pour* lorsqu'elles étaient manquantes. Dans les cas d'alternance en *à* ou en *de*, il a fallu à chaque fois déterminer la fonction syntaxique de l'argument en présence : *Objà*/*Objde* (objet indirect) ou *Obl* (oblique). Par ailleurs, dans un certain nombre de cas où un même lemme a plusieurs sens et donc plusieurs entrées dans le Lefff, il nous a fallu identifier celle(s) des entrées qui étaient concernée(s). Pour certains verbes, un changement de l'inventaire d'entrées s'est avéré nécessaire.

Après ce travail, et outre les nombreuses améliorations et corrections réalisées en cours de route, la couverture du Lefff en ce qui concerne les arguments en *pour* a considérablement augmenté. Ainsi, le nombre d'entrées verbales admettant un argument en *pour* est passé de 116 à 269. Pour les noms, l'augmentation est de 5 à 84, et de 46 à 53 pour les adjectifs. Les 4 adverbes concernés étaient déjà correctement renseignés dans le Lefff. Outre l'ajout de ces réalisations en *pour*, les alternances en *contre*, *à*, *de* et autres prépositions (y compris la préposition vide) ont été renseignées, ainsi que les informations de contrôle. La version du Lefff ainsi obtenue est numérotée 3.4.

5 Conclusion et perspectives

Nous avons présenté une étude de la sous-catégorisation en *pour* en français, question insuffisamment étudiée jusqu'à présent, notamment en raison de la difficulté qu'il y a à trouver des critères pertinents pour distinguer arguments et modifieurs introduits par cette préposition. La perspective discursive nous a été ici utile, les occurrences de *pour* introduisant des infinitives modifieurs étant des connecteurs de discours. Nous avons montré brièvement comment nous avons complété le Lefff grâce au résultat de ce travail, qui s'appuyait sur les ressources lexicales existantes et sur une annotation manuelle de tous les *pour* introduisant une infinitive dans le French TreeBank.

Les perspectives de ce travail sont doubles. Tout d'abord, étant donnée l'homogénéité sémantique d'une grande partie des lexèmes concernés, nous comptons nous appuyer sur le wordnet WOLF pour extraire de nouvelles entrées susceptibles de sous-catégoriser des arguments en *pour*, à partir de leur proximité lexico-sémantique par rapport à des lexèmes déjà identifiés comme tels. Ensuite, nous souhaitons obtenir des résultats comparatifs en analyse syntaxique symbolique avant et après intégration au Lefff des résultats de cette étude, en nous appuyant sur l'analyseur syntaxique FRMG (Villemonte De La Clergerie, 2013). Il est possible toutefois que ce travail nécessite de modifier les corpus annotés impliqués, dans la mesure où l'annotation des occurrences de *pour* n'est pas toujours cohérente avec les résultats du travail présenté ici : de nombreux arguments en *pour* sont annotés comme modifieurs. Une incorporation de nos annotations manuelles sur les *pour V-inf* et une extension de ces annotations aux autres occurrences de *pour* pourrait s'avérer un préalable nécessaire.

Références

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

CADIOT P. (1990a). Contrôle anaphorique et prépositions. *Langages*, **97**, 8–23.

CADIOT P. (1990b). À propos du complément circonstanciel de but. *Langue française*, **86**, 51–64.

COLINET M., DANLOS L., DARGNAT M. & WINTERSTEIN G. (2014). Les emplois de la préposition pour suivie d'une infinitive : description, critères formels et annotation en corpus. In *Actes du 4ème Congrès Mondial de Linguistique Française*, Berlin, Allemagne. À paraître.

COLLINS M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, p. 16–23, Madrid, Spain.

DANLOS L., ANTOLINOS-BASSO D., BRAUD C. & ROZE C. (2012). Vers le FDTB : French Discourse Tree Bank. In *TALN 2012 : 19ème conférence sur le Traitement Automatique des Langues Naturelles*, p. 471–478, Grenoble, France.

DUBOIS J. & DUBOIS-CHARLIER F. (1997). *Les verbes français*. Paris, France : Larousse-Bordas.

GROSS M. (1975). *Méthodes en syntaxe : Régimes des constructions complétives*. Paris, France : Hermann.

HUDDLESTON R. & PULLUM G. (2002). *The Cambridge Grammar of the English Language*. Cambridge : Cambridge University Press.

MIRROSHANDEL S. A., NASR A. & SAGOT B. (2013). Enforcing Subcategorization Constraints in a Parser Using Sub-parses Recombining. In *Proceedings of NAACL 2013*, Atlanta, États-Unis.

RIEZLER S., KING T. H., KAPLAN R. M., CROUCH R., MAXWELL III J. T. & JOHNSON M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, p. 271–278, Philadelphie, Pennsylvanie, États-Unis.

SAGOT B. (2010). The Lefff, a freely available, accurate and large-coverage lexicon for French. In *Proc. of the 7th Language Resource and Evaluation Conference*, La Valette, Malte.

SAGOT B. & DANLOS L. (2012). Merging syntactic lexica : the case for French verbs. In *LREC'12 Workshop on Merging Language Resources*, Istanbul, Turquie.

TOLONE E. (2011). *Analyse syntaxique à l'aide des tables du Lexique-Grammaire du français*. PhD thesis, LIGM, Université Paris-Est, France, Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est Marne-la-Vallée, France.

VAN DEN EYNDE K. & MERTENS P. (2006). Valency dictionary - DICOVALENCY : user's guide. `http://bach.arts.kuleuven.be/dicovalence/`.

VILLEMONTE DE LA CLERGERIE É. (2013). Improving a symbolic parser through partially supervised learning. In *Proceedings of the 13th International Conference on Parsing Technologies (IWPT)*, Naria, Japon.

WYLER G. (2014). Manuel de la grammaire française. `http://gabrielwyler.com/accueil.html`.

Étiquetage morpho-syntaxique pour des mots nouveaux

Ingrid Falk Delphine Bernhard Christophe Gérard Romain Potier-Ferry
LiLPa, Université de Strasbourg
ifalk, dbernhard, christophegerard@unistra.fr ; romainpotierferry@gmail.com

Résumé. Les outils d'étiquetage automatique sont plus ou moins robustes en ce qui concerne l'étiquetage de mots inconnus, non rencontrés dans le corpus d'apprentissage. Il est important de connaître de manière précise la performance de ces outils lorsqu'on cible plus particulièrement l'étiquetage de néologismes formels. En effet, la catégorie grammaticale constitue un critère important à la fois pour leur identification et leur documentation. Nous présentons une évaluation et une comparaison de 7 étiqueteurs morphosyntaxiques du français, à partir d'un corpus issu du Wiktionnaire. Les résultats montrent que l'utilisation de traits de forme ou morphologiques est favorable à l'étiquetage correct des mots nouveaux.

Abstract. Part-of-speech (POS) taggers are more or less robust with respect to the labeling of unknown words not found in the training corpus. It is important to know precisely how these tools perfom when we target part-of-speech tagging for formal neologisms. Indeed, grammatical category is an important criterion for both their identification and documentation. We present an evaluation and comparison of 7 POS taggers for French, based on a corpus built from Wiktionary. The results show that the use of form-related or morphological features supports the accurate tagging of new words.

Mots-clés : étiquetage morphosyntaxique, évaluation, néologie formelle.

Keywords: part-of-speech tagging, evaluation, formal neologisms.

1 Introduction

Les outils d'étiquetage morphosyntaxique procèdent par apprentissage automatique à partir d'un corpus annoté manuellement. De fait, ils sont plus ou moins robustes en ce qui concerne l'étiquetage de mots inconnus, non rencontrés dans le corpus d'apprentissage. Cette problématique nous intéresse tout particulièrement dans le cadre de l'identification de néologismes formels, c'est-à-dire les nouvelles formes qui apparaissent quotidiennement. En effet, la catégorie grammaticale constitue un critère important à la fois pour l'identification et la documentation de ce type de néologismes.

L'objectif de ce travail est de présenter une analyse détaillée de 7 étiqueteurs pour le français, en comparant leurs performances pour l'étiquetage de néologismes issus du Wiktionnaire. L'objectif final est d'identifier l'outil ayant les meilleurs résultats afin de l'incorporer à un système d'identification automatique de néologismes formels. Nous souhaitons avant tout disposer d'un outil qui convient à nos besoins : (i) prêt à l'emploi (ii) sans entraînement supplémentaire (iii) simple d'utilisation et (iv) librement disponible. Cet outil doit par ailleurs avoir les meilleures performances possibles pour l'étiquetage des néologismes dans ces conditions, et sa performance pour les mots connus n'est pas décisive dans le cadre de notre projet.

L'article est organisé comme suit : dans un premier temps, nous présentons le corpus utilisé dans le cadre de nos expériences, puis les outils comparés. La Section 4 détaille le protocole utilisé pour nos expériences et la Section 5 propose une discussion des résultats obtenus.

2 Corpus de référence

Nous avons constitué un corpus de néologismes contextualisés (phrases ou paragraphe), extrait de la liste fournie par le Wiktionnaire [1]. Ce corpus se constitue de 152 entrées [2] toutes conformes au trois critères suivants :

i) Le néologisme et son contexte sont attestés, et non pas forgés.

ii) Il existe au moins une source en format numérique aisément accessible.

iii) La date d'apparition du néologisme est postérieure à 2000.

Techniquement ce corpus se présente sur la forme d'un fichier XML dont un extrait est montré en Figure 1. Les informations utilisés dans nos expériences sont : le lemme du néologisme (balise `word`), sa catégorie grammaticale (attribut `pos`) ainsi que les phrases où il apparaît (balise `texte`). Dans ces phrases le néologisme est marqué par la balise `<neologisme>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<corpus>
  <entry id="L00001">
    <word pos="adv">abracadabrantesquement</word>
    <examples>
      <texte url="http://www.bruxelles-francophone.be/topic2035.html"
             type="forum" date="2009" id="L00001-1">
        Son appétit institutionnel est <neologisme>abracadabrantesquement</neologisme> vorace.
      </texte>
    </examples>
  </entry>
  ....
  <entry id="L00003">
    <word pos="verbe">accessibiliser</word>
    <examples>
      <texte url="http://fplanque.net/Blog/itTrends/2003/01/22/accessibiliser_l_internet_oui_mais_pas_a"
             date="2003" type="blog" id="L00003-1">
        <neologisme>Accessibiliser</neologisme> l'Internet... oui, mais pas à moitié!
      </texte>
      <texte url="http://blog.temesis.com/post/2009/10/25/Marcel-fais-peter-ton-tableau"
             date="2009" type="blog" id="L00003-2">
        Ne pas passer une journée à tenter d'<neologisme>accessibiliser</neologisme>
        deux-trois tableaux hallucinants, pour un gain trop faible.
      </texte>
    </examples>
  </entry>
  ...
```

FIGURE 1: Extrait du corpus de néologismes utilisé dans nos expériences.

La Table 1 résume quelques propriétés de notre corpus.

#	total	noms	verbes	adj	adverbes	locutions	mots à tiret
lemmes	158 [3]	84	36	29	3	6	11
occurences	459	293	68	81	4	13	28
phrases	220	115	55	38	4	8	19

TABLE 1: Propriétés du corpus de néologismes.

1. http://fr.wiktionary.org/wiki/Cat\%E9gorie:N\%E9ologismes_en_fran\%C3\%A7ais

2. Signalons toutefois qu'il est possible que certaines de ces entrées se trouvent déjà dans des ressources lexicales plus récentes. Ainsi le mot *assistanat* est présent dans le Lefff (Sagot, 2010) et *animaliste* peut être trouvé dans le lexique Wortschatz (Quasthoff *et al.*, 2006). Nous estimons néanmoins que cette collection présente un échantillon de formes assez récentes et par conséquence assez représentatif pour permettre une évaluation de l'étiquetage automatique appliquée à de nouveaux mots.

3. Le nombre de formes distinctes est bien de 152, mais nous comptons plusieurs fois celles qui ont plusieurs catégories grammaticales (par exemple animaliste, nom et animaliste, adjective.)

3 Présentation des étiqueteurs comparés

Pour notre projet de collecte de néologismes formels, nous devrons utiliser les étiqueteurs quotidiennement, sur des textes journalistiques.

Les logiciels que nous avons pu identifier et qui répondent aux conditions énoncées dans l'introduction sont présentés brièvement dans cette section. Il s'agit de LGtagger et SEM (Constant & Sigogne, 2011; Constant *et al.*, 2011), LIA_tagg (Nasr *et al.*, 2004), l'étiqueteur de Stanford (Toutanova *et al.*, 2003), MElt (Denis & Sagot, 2010), Talismane (Urieli & Tanguy, 2013) et le plus ancien mais encore largement utilisé, le TreeTagger (Schmid, 1994). Ces outils se distinguent non seulement par les algorithmes appliqués mais leurs performances dépendent aussi d'autres facteurs, comme par exemple :

– le corpus d'apprentissage (et le jeu d'étiquettes utilisées) ;
– des ressources lexicales externes utilisées ou non ;
– du pré- et/ou post traitement.

A leur tour, les algorithmes d'apprentissage et d'étiquetage s'appuient largement sur un ensemble de traits extraits du corpus d'apprentissage, qui ont aussi un effet important sur leur performance.

LGtagger et SEM (Constant & Sigogne, 2011; Constant *et al.*, 2011) LGTagger et SEM ont un fonctionnement similaire : ils procèdent par apprentissage automatique avec des CRF (*Conditional Random Fields* / champs markoviens conditionnels) et peuvent exploiter des lexiques externes. Ces outils visent également à combiner segmentation en unités et étiquetage, en reconnaissant les unités polylexicales. Les deux outils diffèrent toutefois par les attributs et ressources externes utilisées. L'exactitude (*accuracy*) de SEM varie de 81,6% (corpus oral) à 95.6% (corpus blog) [4]. Celle de LGTagger est de 97.7% (sans segmentation incorporée). Nous avons utilisé la version 1.1 de LGTagger, qui intègre plusieurs ressources lexicales (cf. Tableau 2) et où la segmentation est incorporée. SEM a été utilisé avec un modèle appris sur un corpus ne contenant pas d'unités multi-mots et en prenant en compte des ressources linguistiques externes (le Lefff). Comme pour LGTagger, la segmentation était incorporée.

LIA_tagg (Nasr *et al.*, 2004) LIA_tagg est un étiqueteur morpho-syntaxique pour le français prêt à l'emploi développé au Laboratoire Informatique d'Avignon. Il utilise un jeu de 103 étiquettes et implémente une approche probabiliste basée sur les Chaînes de Markov Cachées (HMM). Nous n'avons pas plus d'information sur le corpus d'apprentissage et le taux de précision obtenu.

Stanford (Toutanova *et al.*, 2003) Le *Stanford tagger* est un étiqueteur développé en 2003 pour l'anglais. Ils a été étendu à d'autre langues (français, arabe, chinois, allemand ...), constamment amélioré et est distribué librement à `http://nlp.stanford.edu/software/tagger.shtml#Download`. Sa particularité est d'appliquer un modèle markovien *bidirectionel* à maximisation d'entropie et d'utiliser des propriétés morphologiques : *n*-grammes de suffixes et préfixes, présence de tirets, majuscules, etc. Nous utilisons le modèle français entrainé sur la French Treebank (FTB, (Abeillé *et al.*, 2003)) avec un jeu de 15 étiquettes. Nous n'avons pas d'informations sur la performance de ce modèle.

MElt (Denis & Sagot, 2010) MElt est un étiqueteur basé sur un modèle markovien à maximisation d'entropie qui intègre également un lexique exogène à large couverture (le Lefff (Sagot, 2010)). La méthode d'étiquetage et les traits utilisés sont similaires à ceux de l'étiqueteur de Stanford (cf. Section 3). Dans nos expériences nous avons utilisé la version librement disponible à `http://gforge.inria.fr/projects/lingwb/` qui a été entrainé sur la French Treebank (FTB, (Abeillé *et al.*, 2003)) avec un jeu de 29 étiquettes. Evalué sur le FTB, MElt atteint un taux de précision de 97.75% (91.36% sur les mots inconnus).

Talismane (Urieli & Tanguy, 2013) Talismane est un analyseur syntaxique qui intègre les différentes étapes d'analyse : segmentation en mots et en phrases, étiquetage morphosyntaxique et finalement repérage des dépendances syntaxiques. Pour chaque module, un modèle est appris à partir d'un corpus d'entraînement (FTB). Au moment de l'analyse, des règles peuvent être appliquées pour contraindre les résultats. L'exactitude de Talismane pour le FTB est de 97,8%.

4. `http://www.lattice.cnrs.fr/sites/itellier/SEM.html`

Outil	Méthode	Corpus d'apprentissage (étiquettes)	Ressources lexicales	Utilisation de la forme	Particularité
LGTagger	CRF (i)	FTB (ii)	DELA, Lefff, Prolex, Organisations, Prénoms	Oui (iv)	Segmentation incorporée
SEM	CRF (i)	FTB (ii)	Lefff	Oui (iv)	
LIA_tagg	HMM (iii)	103	lexique 10 000 mots	Non	
Stanford	CMM à maximisation d'entropie.	FTB (ii, 14 étiquettes [6])	–	Oui (iv)	bidirectionnel
MElt	comme *stanford*	FTB (ii, 29 étiquettes)	Lefff	Oui (iv)	
Talismane	classifieur par entropie maximale	FTB	Lefff	Oui	
TreeTagger	Arbres de décisions	43 834 mots, 33 étiquettes	–	Oui (v)	

TABLE 2: Les étiqueteurs morpho-syntaxique utilisés.

(i) CRF : Conditional Random Fields (Champs Aléatoires Conditionnels)

(ii) FTB : French Treebank (Abeillé *et al.*, 2003)

(iii) HMM : Hidden Markov Model (Modèle de Markov caché)

(iv) Utilisation de traits « mots inconnus » : *n*-gram suffixes et préfixes (*n* de 1 à 4), tirets, majuscules, chiffres ...

(v) Utilise un lexique associant à des suffixes la probabilité des étiquettes.

TreeTagger (Schmid, 1994) Le TreeTagger est un des plus anciens outils d'étiquetage automatique. Il est basé sur un modèle probabiliste d'arbre de décision. La version (française) que nous avons utilisé est un modèle issu d'un apprentissage sur un corpus d'environ 44 000 mots et un jeu de 33 étiquettes réalisé par Achim Stein [5]. Dans nos expériences nous n'avons pas utilisé de ressource lexicale supplémentaire.

Le Tableau 2 présente un résumé comparatif des outils employés dans nos expériences.

4 Expériences

Afin d'évaluer l'étiquetage automatique pour la détection et la documentation de néologismes formels, nous appliquons les étiqueteurs décrits en Section 3 au corpus de néologismes que nous avons constitué (présenté en Section 2). Nous comparons ensuite l'étiquetage des outils à celui du corpus de référence.

Étiquettes. Les néologismes du corpus n'ayant que les catégories nom, verbe, adjectif et adverbe, nous projetons les sorties des étiqueteurs, qui sont souvent plus détaillés, sur ces catégories principales. Le nombre d'étiquettes ayant un effet important sur la qualité de l'étiquetage, ce choix d'étiquettes risque de pénaliser les outils produisant un étiquetage plus détaillé. Nous considérons l'étiquetage d'une locution comme correct si les étiquettes des composantes correspondent à celles de la référence.

Prétraitement. Le prétraitement, plus spécifiquement la segmentation en mots, est une étape décisive dans l'étiquetage automatique et chaque étiqueteur a sa propre stratégie pour l'aborder. En général, les outils distribués sont assortis de scripts pour d'abord découper le texte en phrases et ensuite en unités de formes (des mots), qui convient plus ou moins à notre format d'entrée. Pour d'autres outils, comme LGtagger et SEM, la segmentation est intégrée à l'étiquetage.

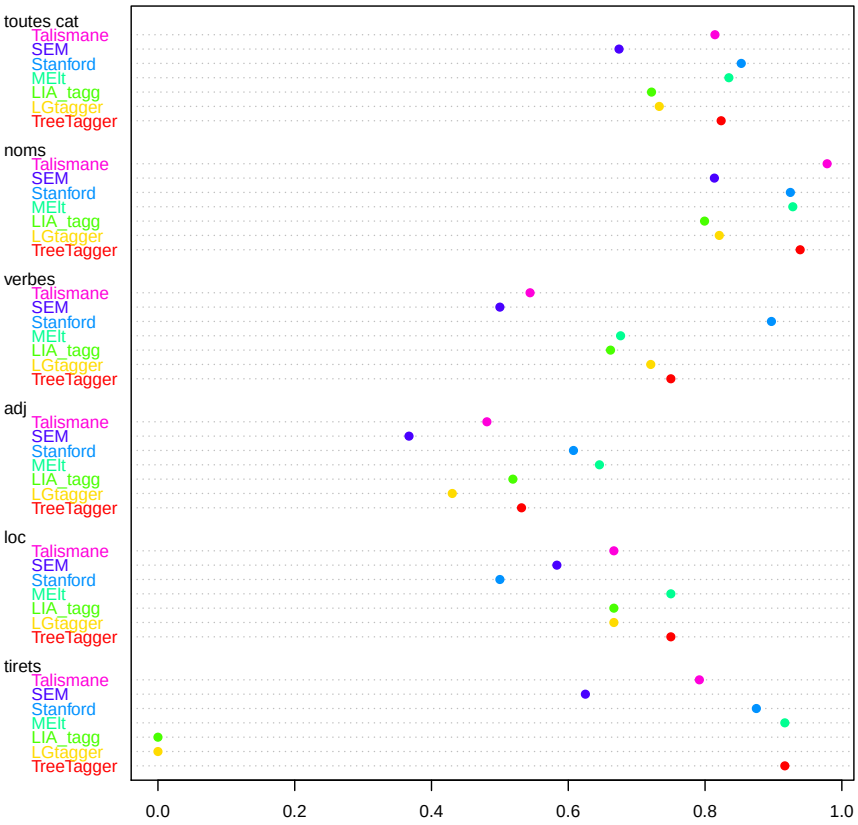
5. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

6. Si la FTB utilise un ensemble de 15 étiquettes morpho-syntaxiques, nous avons constaté que le Stanford Tagger ne distingue pas les noms propres (NP) des noms communs (NC).

5 Résultats et discussion

Étiqueteur	toutes	noms (293)	verbes (68)	adjectifs (81)	locutions (13)	mots à tiret (28)
LGtagger	73.30	82.08	72.06	43.04	66.67	0.00
LIA_tagg	72.17	79.93	66.18	51.90	66.67	0.00
MElt	83.26	92.83	67.65	64.56	75.00	91.67
SEM	67.42	81.36	50.00	36.71	58.33	62.50
Stanford	85.29	92.47	89.71	60.76	50.00	87.50
Talismane	81.45	97.85	54.41	48.10	66.67	79.17
TreeTagger	82.35	93.91	75.00	53.16	75.00	91.67
majorité	86.43					

(a) Pourcentages d'étiquettes correctes.



(b) Représentation graphique.

FIGURE 2: Résultats par catégorie grammaticales.

Les résultats sont présentés en Figure 2. Nous montrons le pourcentage d'étiquettes correctes pour toutes les occurrences de mots nouveaux dans notre corpus et également pour les catégories principales (nom, verbe, adjectif). À titre indicatif est affiché également le pourcentage d'étiquetages correctes pour les locutions mais nous considérons que la taille de cet échantillon ne permet pas de conclusions finales. Les néologismes formels sont souvent formés par composition avec des tirets, pour cette raison nous considérons également les performances des étiqueteurs sur ce type de phénomène [7].

Les performances observées sont, d'une manière générale, plus basses que celles rapportés dans des publications. Ceci

7. A titre indicatif, l'échantillon étant trop limité.

est un effet attendu, vu que les performances publiées sont calculées sur un corpus test issu du même corpus global. Les meilleurs étiquetages sont ceux du Stanford tagger (85.29%) suivis de ceux du MElt (83.26%) et du TreeTagger (82.35%). En mutualisant ces résultats par un vote majoritaire le taux a pu être augmenté à 86.43%.

Par rapport à l'algorithme d'étiquetage, les CMM utilisés pour Stanford et MElt semblent plus favorables dans le cadre de notre application que les CRF (employés par LGTagger et SEM) ou les classifieurs à entropie maximale.

Pour ce qui est de ressources lexicales, leur emploi n'a pas d'effet positif dans notre cas. En effet, alors que Stanford et MElt s'appuient sur des algorithmes et traits similaires, l'utilisation d'une ressource lexicale externe par MElt ne lui permet pas d'améliorer ses performances. Par contre, l'utilisation de traits de forme ou morphologiques (préfixe, suffixe, etc.) semble favorable : Stanford, MElt et TreeTagger, qui les intégrent obtiennent les meilleurs taux d'étiquetage correctes. Il a par ailleurs été montré que ces traits jouent un rôle important pour l'identification de néologismes formels (Sagot *et al.*, 2013; Falk *et al.*, 2014).

A son tour LIA_tagg semble défavorisé par le grand nombre d'étiquettes utilisées.

6 Conclusion

Au vu de ces résultats, nous concluons que le *Stanford Tagger* est l'outil le plus plus adapté à notre application et nous allons donc l'utiliser dans notre projet. Comme c'était à prévoir l'utilisation de ressources lexicales externes n'a pas d'impact. Par ailleurs, la combinaison de la segmentation et de l'étiquetage est plutôt défavorable. Par contre les traits de forme et morphologiques ont un impact positif.

Remerciements. Ces travaux ont été financés par l'Université de Strasbourg dans le cadre de l'Initiative d'Excellence (IdEx) 2012 (projet Logoscope).

Références

ABEILLÉ A., CLÉMENT L. & TOUSSENEL F. (2003). Building a Treebank for French. In A. ABEILLÉ, Ed., *Treebanks*, number 20 in Text, Speech and Language Technology.

CONSTANT M. & SIGOGNE A. (2011). MWU-aware part-of-speech tagging with a CRF model and lexical resources. In *Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World, ACL 2011*.

CONSTANT M., TELLIER I., DUCHIER D., DUPONT Y., SIGOGNE A. & BILLOT S. (2011). Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français. In *Actes de TALN 2011*.

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

FALK I., BERNHARD D. & GÉRARD C. (2014). From Non Word to New Word : Automatically Identifying Neologisms in French Newspapers. In *Proceedings of LREC 2014*, Reykjavik, Islande.

NASR A., BÉCHET F. & VOLANSCHI A. (2004). Tagging with Hidden Markov Models Using Ambiguous Tags. In *Proceedings of COLING 2004*, COLING '04, Stroudsburg, PA, USA.

QUASTHOFF U., RICHTER M. & BIEMANN C. (2006). Corpus Portal for Search in Monolingual Corpora. In *Proceedings of LREC 2006*, Genoa.

SAGOT B. (2010). The Lefff, a Freely Available and Large-coverage Morphological and Syntactic Lexicon for French. In *Proceedings of LREC'10*, Valletta, Malta.

SAGOT B., NOUVEL D., MOUILLERON V. & BARANES M. (2013). Extension dynamique de lexiques morphologiques pour le français à partir d'un flux textuel. In *Actes de TALN 2013*.

SCHMID H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

TOUTANOVA K., KLEIN D., MANNING C. D. & SINGER Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of ACL 2003*.

URIELI A. & TANGUY L. (2013). L'apport du faisceau dans l'analyse syntaxique en dépendances par transitions : études de cas avec l'analyseur talismane. In *Actes de TALN 2013*.

Détection et correction automatique d'entités nommées dans des corpus OCRisés

Benoît Sagot Kata Gábor
Alpage, INRIA & Université Paris-Diderot, 75013 Paris
{prenom.nom}@inria.fr

Résumé. La correction de données textuelles obtenues par reconnaissance optique de caractères (OCR) pour atteindre une qualité éditoriale reste aujourd'hui une tâche coûteuse, car elle implique toujours une intervention humaine. La détection et la correction automatiques d'erreurs à l'aide de modèles statistiques ne permettent de traiter de façon utile que les erreurs relevant de la langue générale. C'est pourtant dans certaines entités nommées que résident les erreurs les plus nombreuses, surtout dans des données telles que des corpus de brevets ou des textes juridiques. Dans cet article, nous proposons une architecture d'identification et de correction par règles d'un large éventail d'entités nommées (non compris les noms propres). Nous montrons que notre architecture permet d'atteindre un bon rappel et une excellente précision en correction, ce qui permet de traiter des fautes difficiles à traiter par les approches statistiques usuelles.

Abstract. Correction of textual data obtained by optical character recognition (OCR) for reaching editorial quality is an expensive task, as it still involves human intervention. The coverage of statistical models for automated error detection and correction is inherently limited to errors that resort to general language. However, a large amount of errors reside in domain-specific named entities, especially when dealing with data such as patent corpora or legal texts. In this paper, we propose a rule-based architecture for the identification and correction of a wide range of named entities (proper names not included). We show that our architecture achieves a good recall and an excellent correction accuracy on error types that are difficult to adress with statistical approaches.

Mots-clés : OCR, Entités nommées, Détection d'erreurs, Correction d'erreurs.

Keywords: OCR, Named entities, Error Detection, Error Correction.

1 Introduction et état de l'art

Le projet dans lequel s'inscrit le travail présenté ici a pour objectif d'optimiser des performances des projets de numérisation patrimoniale par le biais de l'automatisation de la détection et de la correction d'erreurs dans des documents en sortie de systèmes de reconnaissance optique de caractères (Optical Character Recognition, ou OCR). En effet, la correction de données textuelles obtenues par reconnaissance optique de caractères (OCR) pour obtenir une qualité éditoriale est aujourd'hui une tâche coûteuse, même lorsque les documents d'origine sont d'excellente qualité, car elle implique toujours une intervention humaine, seule à même de garantir des taux d'erreurs acceptables par exemple pour publication sous forme de livre électronique. Pour diminuer le coût de cette intervention humaine, des approches automatiques pour la détection d'erreurs, pour la suggestion de corrections voire pour la correction automatique sont progressivement utilisées. Ces approches reposent presque toujours sur l'utilisation de modèles de langage et de modèles d'erreur, mais rarement sur des informations linguistiques plus riches. Pourtant, ces techniques ne permettent pas de traiter de façon utile les erreurs qui ne relèvent pas de la langue générale, et notamment les erreurs présentes dans les entités nommées. C'est pourtant dans certains types d'entités nommées que résident les erreurs les plus nombreuses, surtout dans des données telles que des corpus de brevets (formules chimiques) ou des textes juridiques (identifiants de jurisprudence). En effet, de telles entités ne peuvent être présentes dans les lexiques sur lesquels reposent les systèmes d'OCR. Elles ne sont pas adaptées à des traitements statistiques, et sont au contraire adaptées à une modélisation par règles qui en permette la détection robuste, malgré le bruit issu de l'OCR, et la correction. Dans cet article, nous proposons ainsi d'étudier l'impact sur la correction automatique de sorties d'OCR d'une prise en charge par règles des erreurs présentes dans un large éventail d'entités nommées.

Les premières approches à la correction automatique des erreurs (orthographiques, lexicales ou post-OCR) s'appuyaient fortement sur le lexique : pour chaque mot inconnu, elles cherchaient des candidats proches (par exemple en termes de

distance d'édition (Levenshtein, 1965)) qui figurent dans le lexique et choisissent en prenant en compte la fréquence des candidats, le contexte, ou éventuellement un poids associé au type d'erreur présumé. Cependant, l'utilisation des lexiques pré-existants semble laisser la place à d'autres méthodes plus flexibles qui ne nécessitent pas la consultation d'une telle ressource, ou qui la créent à la volée à partir du corpus à corriger (Reynaert, 2004) ou en exploitant les données du Web (Cucerzan & Brill, 2004; Strohmaier *et al.*, 2003). Deux méthodes fondamentales peuvent être distinguées dans les travaux plus récents visant la correction des documents bruités : les systèmes s'appuyant sur des automates à états finis (pondérés) (Ringlstetter *et al.*, 2006), et des méthodes adoptant le modèle du canal bruité (Kernighan *et al.*, 1990; Brill & Moore, 2000; Kolak & Resnik, 2002). Contrairement aux systèmes symboliques, ces dernières ne font aucune hypothèse a priori sur le type d'erreurs que l'on prévoit de rencontrer dans le document et se prêtent ainsi plus facilement à l'adaptation à de nouveaux domaines. Cependant, comme la matrice de confusion qui sert de modèle d'erreur est construite à partir d'un corpus, il est souhaitable de disposer d'un corpus d'apprentissage de taille conséquente. Les méthodes symboliques évitent le problème du besoin de ressources annotées et permettent une meilleure intégration d'informations résultant d'une analyse linguistique, mais elles présentent l'inconvénient d'être spécifiques à une langue et, typiquement, à un domaine. D'autres méthodes utilisent une combinaison des sorties de plusieurs OCR : Klein & Kope (2002) utilisent un système de vote, alors que Lund & Ringger (2009) produisent un treillis de mots à partir d'un alignement textuel effectué sur les sorties et choisissent l'une des hypothèses par recherche dans un dictionnaire. Mais ces techniques sont mal adaptées à la correction d'erreurs au sein d'entités nommées souvent complexes, dispersées (*sparse*) et spécifiques au domaine.

Pour palier cette limitation, nous avons adapté et étendu la chaîne de traitement de surface SXPipe (Sagot & Boullier, 2008) pour effectuer la tokenisation, la segmentation en énoncés et le traitement de divers types d'entités nommées dans des sorties d'OCR. En particulier, nous avons développé de nouvelles grammaires locales pour traiter des classes d'entités nommées rencontrées dans les corpus traités dans le projet (formules chimiques, identifiants juridiques), nous avons adapté l'ensemble des grammaires locales pour les rendre robustes aux erreurs d'OCR, et nous avons intégré un nouveau mécanisme permettant de compléter la reconnaissance d'une entité par une proposition de correction selon des règles qui dépendent en partie du type d'entité. L'objectif, à terme, est de coupler cette chaîne avec une architecture plus classique pour la correction des tokens hors-entités (modèle de langage, modèle d'erreur).

Dans ce qui suit, nous allons présenter les corpus traités dans le cadre de nos travaux (section 2). Une description des grammaires locales construites pour la reconnaissance et la correction des entités nommées est fournie à la section 3, et les résultats sont décrits à la section 4.

2 Corpus à traiter

Les trois corpus sur lesquels nous avons travaillé sont présentés à la table 1. Contrairement aux deux autres, le corpus BNF/Gallica est un *corpus d'écarts*, c'est-à-dire un corpus constitué d'une version brute (sortie directe d'OCR) et d'une version corrigée par des opérateurs humains (qualité éditoriale), les deux versions étant alignées. Les corpus d'écarts contribuent à créer des modèles d'erreur.

Corpus	**Genre**	**Nb. de tokens**	**Type de corpus**
BNF/Gallica	littérature	50 000 000	corpus d'écarts
OPOCE	jurisprudence	37 000 000	sortie d'OCR
EPO (European Patent Office)	brevets	15 000 000	sortie d'OCR

TABLE 1 – Corpus utilisés dans cette étude

L'étude de ces corpus nous a permis d'identifier certains types d'entités comportant de nombreuses erreurs d'OCR. Ainsi, dans le corpus de brevets EPO, une proportion importante des erreurs d'OCR sont situées au sein de formules chimiques, indications de mesures et numéros de brevets. Le corpus de jurisprudence européenne contient de nombreux identifiants juridiques tels que des références à des articles de lois, des arrêtés ou des décisions.

De manière générale, pour ces entités comme pour les entités plus classiques (dates, adresses…), qui contiennent également des erreurs d'OCR que nous souhaitons corriger, une approche par règles semble plus adaptée qu'une approche par modèle de langue, en raison de la grande variété des occurrences de ces entités, mais également en raison de leur forte structuration et des nombreuses contraintes non-linguistiques qui s'y appliquent.

3 Grammaires locales pour la correction de sorties d'OCR

3.1 Définition des entités nommées

La tâche d'étiquetage des entités nommées a reçu une attention considérable au sein de la communauté TAL depuis les années 90. Une des tâches partagées de la série de conférences MUC (*Message Understanding Conference* (Chinchor, 1998)) avait pour objectif de reconnaître les entités nommées, définies en tant qu'unités faisant référence à une entité unique et concrète et réalisées par des noms propres (noms de personnes, d'organisations, d'artefacts ou de lieux). Les expressions temporelles et les expressions de quantité sont généralement ajoutées à cette liste, moins en raison de leurs propriétés sémantiques que pour des considérations d'ordre pratique. Toutefois, la plupart des campagnes d'évaluation (p.ex. CoNLL 2003, (Sang & Meulder, 2003)) se contentent d'illustrer les différentes catégories d'entités nommées de façon extensionnelle, par des exemples.

Dans cet article nous définissions une (mention d')entité nommée comme une séquence de caractères ou de tokens qui n'est analysable ni morphologiquement ni syntaxiquement, mais procède d'un motif productif, qui peut contenir des marques de ponctuation. Nous utilisons ainsi le terme d'entité nommée dans un sens légèrement plus large que le sens habituel (Maynard *et al.*, 2001), en y incluant également les nombres, les formules chimiques, les unités monétaires, etc. En revanche, nous ne traitons pas ici des entités nommées classiques que sont les noms de personnes, de lieux ou d'organisations (sauf en tant que composants d'entités plus larges comme les identifiants juridiques).

3.2 La chaîne de traitement SXPipe-postOCR

SXPipe (Sagot & Boullier, 2008) est une chaîne de traitement dont le rôle est d'appliquer à des corpus une cascade de traitements linguistiques de surface. Selon leur fonction, ces traitements peuvent être classés en trois catégories : détection de frontières (entre phrases ou mots), correction d'erreurs et reconnaissance des entités nommées. Tous les modules de la chaîne sont paramétrables, pour s'adapter à différentes applications. De plus, chaque utilisateur peut construire sa propre chaîne de traitement avec ses propres paramètres et jeux de modules, mais également créer et intégrer ses propres modules. Nous avons ainsi construit un exécutable spécifique SXPipe-postOCR, qui repose sur des versions améliorées de modules existants et sur de nouveaux modules, auxquels ont été ajoutés des règles permettant de proposer une correction pour certaines entités reconnues.

SXPipe contenait déjà des grammaires locales effectuant la reconnaissance de nombreux types d'entités nommées, comme les expressions temporelles ou les adresses. Elles ont été améliorées à la fois en couverture et en robustesse, afin de reconnaitre ces entités même dans des données légèrement bruitées issues de systèmes d'OCR. De plus, de nouvelles grammaires locales ont été développées spécifiquement pour ce projet, notamment pour la reconnaissance des dimensions, des unités monétaires, des formules chimiques et des identifiants de jurisprudence, toujours avec un souci de large couverture et de robustesse. Enfin, ces grammaires ont été complétées par des règles de *correction*, dont l'application (facultative) permet d'obtenir pour certaines entités nommées reconnues une version normalisée/corrigée. À cet égard, par prudence, l'accent a été mis sur la précision des propositions de correction, une correction incorrecte étant bien plus dommageable qu'une absence de correction pour ce type de tâche. Ainsi, la quasi-totalité des grammaires se basent sur la présence de marqueurs non-ambigüs, et une grande partie d'entre elles exploitent de l'information contextuelle.

3.2.1 Détection robuste et correction automatique dans les grammaires locales de SXPipe

La reconnaissance et la correction/normalisation des entités sont assurées par ces grammaires locales (cascades d'expressions régulières), chaque type d'entité correspondant à un module distinct. Les propositions de correction/normalisation bénéficient ainsi de connaissances spécifiques sur leur nature et leur structure interne. Par exemple, dans le cas de dimensions (p.ex. : 10^3 l/m^2), la structure en valeur et/ou puissance de 10 puis unité de mesure permet d'appliquer des corrections spécifiques : la puissance de 10 peut être mise en exposant, et l'unité de mesure peut voir tous ses chiffres 1 transformés en lettres l, tous les autres chiffres mis en exposant et tous les espaces supprimés (sous condition). C'est ainsi que l'on peut corriger « 10 3 1/m2 » en « 10^3 l/m^2 ».

L'identification de ces règles de correction a été faite par exploration manuelle du corpus d'écarts BNF/Gallica, à l'exception des formules chimiques. Ce dernier cas appelle par ailleurs quelques précisions supplémentaires (cf. section suivante).

Les tables 2 et 3 illustrent les différents cas possibles (détection ou non-détection, normalisation proposée ou non, correcte

ou non, ayant introduit ou non des erreurs). La table 2 rassemble des exemples pour lesquels cette normalisation est totalement correcte, la table 3 illustre diverses difficultés rencontrées par SXPipe-postOCR.

Sortie d'OCR	**Étiquetage**	**Normalisation proposée = Référence**
23 avril i908 1" janvier 1912 1G février 1859	{23 avril i908} _DATE {1" janvier 1912} _DATE {1G février 1859} _DATE	23 avril 1908 1er janvier 1912 16 février 1859
Al(OH)3 Ba2Na(NbO3)5 C02	{Al(OH)3} _CHEM {Ba2Na(NbO3)5} _CHEM {C02}_CHEM	$Al(OH)_3$ $Ba_2Na(NbO_3)_5$ CO_2
MgA1204	{MgA1204}_CHEM	$MgAl_2O_4$
484,9 g.l-1 2,10 10 6 1/m2	{484,9} _NUM {g.l-1} _DIMENSION {2,10 10 6 1/m2} _DIMENSION	484,9 $g.l^{-1}$ 2,10 10^6 l/m^2
52, rue de Chabrol, Paris, Xèriie rue de l'Université, i3 52, rue Taibout, Paris (9') 1, bd Beau-Séjour, Paris (16°)	{52, rue de Chabrol, Paris, Xèriie} _ADRESSE {rue de l'Université, i3} _ADRESSE {52, rue Taibout, Paris (9')} _ADRESSE {1, bd Beau-Séjour, Paris (16°)} _ADRESSE	52, rue de Chabrol, Paris, Xème rue de l'Université, i3 52, rue Taibout, Paris (9e) 1, bd Beau-Séjour, Paris (16e)

TABLE 2 – Reconnaissance et correction/normalisation des entités bruitées : exemples corrects

Sortie d'OCR	**Étiquetage**	**Normalisation proposée**	**Référence**
15 rue Spufflot 3ÏÏ, rue Taitboùt, Paris	{15 rue Spufflot} _ADRESSE {3ÏÏ, rue Taitboùt, Paris} _ADRESSE	15 rue Spufflot 31, rue Taitboùt, Paris	15 rue Soufflot 31, rue Taitbout, Paris
C8-C20 (CH2)nNR4R5 R1COOH	{C8-C20} _CHEM ({CH2} _CHEM)nNR4R5 R1COOH	C_8-C_2O (CH_2)nNR4R5 R1COOH	C_8-C_{20} $(CH_2)_n NR_4R_5$ R_1COOH

TABLE 3 – Reconnaissance et correction/normalisation des entités avec bruit : autres exemples

3.2.2 Le cas des formules chimiques

Les formules chimiques, malgré les nombreuses régularités et contraintes qui les régissent, représentent un défi important pour la reconnaissance automatique. Compte tenu de leur fréquence dans le corpus de brevets EPO et des nombreuses erreurs d'OCR que l'on y rencontre, nous leur avons consacré une attention particulière. Au lieu d'utiliser un des outils existants (Hawizy *et al.*, 2011; Rocktäschel *et al.*, 2012) pour la reconnaissance des formules, nous avons décidé de créer notre propre grammaire, ce qui permet une meilleure prise en charge des erreurs d'OCR. La grammaire SXPipe dédiée que nous avons développée est donc robuste à certaines erreurs, telles que la substitution de la lettre l par le chiffre 1, de la lettre O par le chiffre 0, et par le fait que les nombres qui devraient être en indices ne le sont pas. Certaines règles de correction exploitent des connaissances chimiques (par exemple, un élément chimique ou une sous-formule ne peut pas être indicé par 0 ou 1). Ainsi, si l'on donne MgA1204 en entrée à ce module, il produit en sortie {MgA1204}_CHEM_$MgAl_2O_4$_. Les tables 2 et 3 donnent quelques autres exemples.

Outre la sutructure interne des formules chimiques, la grammaire s'appuie sur leur contexte d'apparition. Dans la description de la structure des formules potentielles, nous sommes partis d'une liste complète d'éléments chimiques. Certains d'entre eux, tout comme certaines formules simples, sont toutefois ambigus avec des mots de la langue générale (p. ex. Ce, Ne, La). Nous avons donc construit deux ensembles de règles de reconnaissance. Un premier ensemble, fiable, ne reconnaît que des formules qui ne sont pas ambiguës avec des mots de la langue ou des acronymes. Un second ensemble de règles reconnaît de telles séquences ambiguës. Ce n'est que si les règles les plus sûres se sont appliquées avec succès sur une phrase que nous lui appliquons les règles plus ambigües : en effet, nous supposons alors que nous sommes en présence d'un contexte thématique favorisant l'apparition des formules chimiques.

4 Évaluation

Nous avons effectué une évaluation manuelle des grammaires d'entités nommées et des corrections/normalisations, en nous limitant aux dates, adresses et formules chimiques. Pour les deux premiers types d'entités, nous avons effectué cette évaluation à partir du corpus BNF/Gallica ; les formules chimiques ont été évaluées sur le corpus EPO.

Pour l'évaluation de la reconnaissance des dates, nous avons sélectionné aléatoirement 200 phrases parmi un ensemble de phrases susceptibles de contenir au moins une entité nommée de ce type. Pour ce faire, nous avons eu recours à des marqueurs couvrants (mais moins sûrs que nos grammaires) qui permettent de détecter les dates avec un rappel (presque) parfait mais une précision moindre. L'idée sous-jacente est que le nombre de dates qui ne sont pas capturées par nos marqueurs est très faible, de sorte qu'il est possible de calculer sur un tel échantillon non seulement la précision mais également le rappel de notre grammaire de reconnaissance. Nous avons procédé de même pour les adresses.

Pour l'évaluation manuelle de la correction/normalisation des dates et adresses, nous n'avons pas pu nous appuyer sur ces deux ensembles de 200 phrases, dans la mesure ou le pourcentage d'entités pour lesquelles une modification est proposée est trop bas. Nous avons donc évalué manuellement l'intégralité des cas où une modification est proposée par nos grammaires parmi 500 000 phrases du corpus BNF/Gallica, en nous référant si besoin à la version de qualité éditoriale corrigée par des humains.

Pour les formules chimiques, l'absence de marqueurs vraiment couvrants nous a conduit à choisir un échantillon aléatoire de 200 phrases parmi celles où notre grammaire a détecté au moins une formule chimique. Le rappel calculé sur cet échantillon est donc une borne supérieure du rappel réel (c'est pour cette raison qu'il est entre parenthèses dans la table 4). Toutefois, un examen manuel d'un certain nombre de phrases extraites aléatoirement de tout le corpus nous a montré qu'il n'arrivait presque jamais qu'une formule chimique non-détectée soit dans une phrase dans laquelle aucune autre formule chimique ne l'a été. Ceci laisse penser que le rappel réel est très proche du score obtenu.

L'évaluation manuelle de la correction/normalisation des formules chimiques a pu se faire directement sur ces 200 phrases, dans la mesure où une modification est proposée pour plus des trois quarts des formules chimiques trouvées.

Type d'entité	**Reconnaissance**			**Correction/normalisation**		
	Précision	Rappel	#occurrences pour 10^6 tokens	Précision	Rappel	Proportion d'entités corrigées parmi les entités détectées
Dates	0,98	0,97	4713	0,96	–	2%
Adresses	0,83	0,86	269	0,76	–	3%
Formules chimiques	0,91	(0,88)	300	0,95	0,90	72%

TABLE 4 – Evaluation de la reconnaissance des entités nommées et de leur correction/normalisation. Les fréquences sont calculées sur le corpus BNF/Gallica, sauf pour les formules chimiques où le corpus utilisé est EPO

La table 4 donne les résultats de notre évaluation. Elle indique également la fréquence de chaque type d'entité dans son corpus d'évaluation. Pour le calcul des scores de reconnaissance, une entité n'est considérée comme correctement reconnue que si son type et ses deux bornes l'ont été : les reconnaissances partielles sont considérées comme des erreurs.

La précision de la correction/normalisation est définie comme le pourcentage d'entités modifiées par nos grammaires telles que le résultat de cette modification est complètement correct. Ainsi, une correction partielle compte comme une erreur. Ceci explique que la précision obtenue pour les adresses soit plus faible que pour les autres types d'entités, puisqu'elles contiennent presque toujours des noms propres auxquels nos corrections ne s'appliquent pas.

Le rappel est le pourcentage d'entités correctement modifiées parmi celles qui n'étaient pas totalement correctes dans la sortie brute d'OCR. Pour le calcul des scores de correction/normalisation des formules chimiques, toutes les entités ont été prises en compte, qu'elles aient été ou non correctement détectées par SXPipe-postOCR. Ceci nous a permis le calcul du rappel. Cependant, la taille des corpus utilisés pour l'évaluation des dates et des adresses nous a contraint à nous limiter pour ces deux types d'entités aux occurrences reconnues par SXPipe-OCR. Le calcul du rappel est alors impossible.

5 Conclusion et perspectives

Nous avons présenté le composant linguistique d'une architecture pour la correction automatique des erreurs dans des documents numériques obtenus par reconnaissance optique de caractères. Notre système, constitué d'une cascade de grammaires locales implémentées dans l'outil SXPipe, permet la détection robuste ainsi que la correction de certains types d'entités nommées fréquemment rencontrées dans des corpus spécialisés. Contrairement à un modèle statistique d'erreurs, notre méthode permet une meilleure prise en charge des propriétés formelles non-linguistiques de ces entités. De plus, les règles de corrections sont spécifiques à chaque type d'entité, parfois à des types de segments au sein des entités (par exemple, le numéro d'arrondissement dans une adresse). L'évaluation effectuée à la main, en s'appuyant sur

notre corpus d'écart de qualité éditoriale, confirme l'utilité de grammaires locales de reconnaissance et de correction au sein d'une chaîne de correction automatique des sorties de systèmes d'OCR.

La prochaine étape de notre travail est de réaliser effectivement l'intégration de SXPipe-postOCR au sein d'une chaîne complète de correction de sorties d'OCR, c'est-à-dire de coupler nos grammaires locales de reconnaissance et de correction avec un modèle d'erreur et un modèle de langage statistiques. Pour cela, nous envisageons d'utiliser SXPipe-postOCR comme un pré-traitement qui viendra en amont du composant reposant sur ces deux types de modèles statistiques.

Les perspectives scientifiques de ce travail sont doubles. Nous prévoyons tout d'abord de réaliser des expériences avec différents modèles de langage exploitant l'information issue de l'étiquetage en entités nommées. Une autre extension possible notre travail serait de faire en sorte que les grammaires locales puissent, en cas de doute, proposer plus d'une correction, charge au modèle de langage (voire au modèle d'erreurs) de choisir la meilleure.

Références

BRILL E. & MOORE R. C. (2000). An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL'00)*, p. 286–293, Hong Kong.

CHINCHOR N. (1998). MUC-7 named entity task definition. In *Seventh Message Understanding Conference (MUC-7)*, Fairfax, États-Unis.

CUCERZAN S. & BRILL E. (2004). Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*, p. 293–300, Barcelone, Espagne.

HAWIZY L., JESSOP D. M., ADAMS N. & MURRAY-RUST P. (2011). ChemicalTagger : A tool for semantic text-mining in chemistry. *Journal of Cheminformatics*, **3**(17).

KERNIGHAN M., CHURCH K. & W.A. G. (1990). A spelling correction program based on a noisy channel model. In *Proceedings of the 13th conference on Computational linguistics*, p. 205–210, Helsinki, Finlande.

KLEIN S. T. & KOPE M. (2002). A voting system for automatic OCR correction. In *Proceedings of the Workshop On Information Retrieval and OCR : From Converting Content to Grasping Meaning*, p. 1–21, Tampere, Finlande.

KOLAK O. & RESNIK P. (2002). OCR error correction using a noisy channel model. In *Proceedings of the Second International Conference on Human Language Technology Research (HLT'02)*, p. 257–262, San Diego, États-Unis.

LEVENSHTEIN V. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, **10**, 707–710.

LUND W. B. & RINGGER E. K. (2009). Improving optical character recognition through efficient multiple system alignment. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'09)*, p. 231–240, Austin, États-Unis.

MAYNARD D., TABLAN V., URSU C., CUNNINGHAM H. & WILKS Y. (2001). Named entity recognition from diverse text types. In *In Proceedings of the Recent Advances in Natural Language Processing Conference (RANLP'01)*, p. 257–274, Tzigov Chark, Bulgarie.

REYNAERT M. (2004). Multilingual text induced spelling correction. In *Proceedings of the Workshop on Multilingual Linguistic Ressources (MLR'04)*, p. 117–117.

RINGLSTETTER C., SCHULZ K. U., MIHOV S. & LOUKA K. (2006). The same is not the same—postcorrection of alphabet confusion errors in mixed-alphabet ocr recognition. In *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR'05)*, p. 406–410, Séoul, Corée du Sud.

ROCKTÄSCHEL T., WEIDLICH M. & LESER U. (2012). Chemspot : a hybrid system for chemical named entity recognition. *Bioinformatics*, **28**.

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

SANG E. F. T. K. & MEULDER F. D. (2003). Introduction to the CoNLL-2003 shared task : Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 (CONLL'03)*, p. 142–147, Edmonton, Canada.

STROHMAIER C., RINGLSTETTER C., SCHULZ K. & MIHOV S. (2003). Lexical postcorrection of ocr-results : the web as a dynamic secondary dictionary ? In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03)*, p. 1133–1137, Edinburgh, Royaume-Uni.

Méthodes de lissage d'une approche morpho-statistique pour la voyellation automatique des textes arabes

Amine Chennoufi[1] Azzeddine Mazroui[1]

(1) Université Mohamed I / Faculté des Sciences / Département de Mathématiques et Informatiques
Oujda, Maroc
chennoufi.amin@gmail.com azze.mazroui@gmail.com

Résumé. Nous présentons dans ce travail un nouveau système de voyellation automatique des textes arabes en utilisant trois étapes. Durant la première phase, nous avons intégré une base de données lexicale contenant les mots les plus fréquents de la langue arabe avec l'analyseur morphologique AlKhalil Morpho Sys pour fournir les voyellations possibles pour chaque mot. Le second module dont l'objectif est d'éliminer l'ambiguïté repose sur une approche statistique dont l'apprentissage a été effectué sur un corpus constitué de textes de livres arabes et utilisant les modèles de Markov cachés (HMM) où les mots non voyellés représentent les états observés et les mots voyellés sont ses états cachés. Le système utilise les techniques de lissage pour contourner le problème des transitions des mots absentes et l'algorithme de Viterbi pour sélectionner la solution optimale. La troisième étape utilise un modèle HMM basé sur les caractères pour traiter le cas des mots non analysés.

Abstract. We present in this work a new approach for the Automatic diacritization for Arabic texts using three stages. During the first phase, we integrated a lexical database containing the most frequent words of Arabic with morphological analysis by Alkhalil Morpho Sys which provided possible diacritization for each word. The objective of the second module is to eliminate the ambiguity using a statistical approach in which the learning phase was performed on a corpus composed by several Arabic books. This approach uses the hidden Markov models (HMM) with Arabic unvowelized words taken as observed states and vowelized words are considered as hidden states. The system uses smoothing techniques to circumvent the problem of unseen word transitions in the corpus and the Viterbi algorithm to select the optimal solution. The third step uses a HMM model based on the characters to deal with the case of unanalyzed words.

Mots-clés : Langue arabe, voyellation automatique, analyse morphologique, modèle de Markov caché, corpus, lissage, algorithme de Viterbi.

Keywords: Arabic language**,** Automatic diacritization, morphological analysis, hidden Markov model, corpus, smoothing, Viterbi Algorithm.

1 Introduction

Le système d'écriture arabe est caractérisé dans la plupart des textes par l'absence de signes diacritiques : les voyelles courtes i.e. ◌َ (*fatha*), ◌ُ(*damma*) et ◌ِ (*kasra*), en plus des signes ◌ً, ◌ٍ et ◌ٌ (*tanween*), ◌ّ(*chadda*) et ◌ْ(*sokoun*). L'absence de ces signes engendre une augmentation significative de l'ambiguïté dans le texte arabe, qui peut causer de la confusion dans plus de 90% des mots du texte (Debili, Achour, 1998). Malgré le fait que le lecteur ayant un certain niveau de connaissances de la langue arabe peut facilement récupérer les signes diacritiques absents du texte en se basant sur le contexte des mots et ses connaissances de la morphologie et de la syntaxe de la langue arabe, les textes sans signes diacritiques demeurent un obstacle pour les apprenants non natifs de la langue arabe et les personnes ayant des difficultés d'apprentissage. De même, les limites des performances de plusieurs applications du traitement automatique de la langue arabe (TALA) telles que l'analyse syntaxique, la traduction automatique et les corpus arborés sont dues en partie à l'absence des signes diacritiques dans les textes arabes (Maamouri et al., 2006). En effet, contrairement aux langues européennes où il est facile d'identifier les corresponds en phonèmes oraux des sections écrites (*Text to Speech*), il est impératif pour les textes arabes de récupérer les signes diacritiques avant de procéder à la recherche de leurs correspondants (Vergyri et al,. 2004). Aussi, certaines recherches ont suggéré l'importance d'utiliser les textes voyellés pour accroître l'efficacité des systèmes de reconnaissance de la parole (Messaoudi et al., 2004).

D'autre part, les corpus utilisés dans les modèles de langage ne peuvent pas couvrir tout le vocabulaire. De ce fait, les séquences de mots peuvent avoir une probabilité nulle ce qui peut générer de mauvais résultats. Cet ainsi que des chercheurs ont eu recours aux méthodes de lissage pour contourner ce problème. (Chen, Goodman, 1998) affirment que les techniques de lissage sont essentielle dans la construction des systèmes de reconnaissance de la parole. De même (Manning, Schütze, 1999) précisent que le lissage donnent des performances intéressantes dans les modèles de langage.

Ce papier est organisé comme suit : la deuxième section est réservée à l'état de l'art, la troisième à la présentation de l'approche où nous détaillons au début le fonctionnement de l'analyseur morphologique utilisé dans la première partie de notre système, puis nous rappelons les étapes statistiques adoptées dans la deuxième et la troisième phase de notre méthode. La quatrième section est consacrée aux étapes d'apprentissage et de test du voyelliseur. Enfin, la dernière partie sera consacrée à la conclusion et aux perspectives futures.

2 Etat de l'art

En se référant aux travaux de recherches antérieurs, nous pouvons diviser les tentatives de voyellation automatique des textes arabes en trois parties: approches fondées sur des règles, approches statistiques et approches hybrides.

2.1 Approches fondées sur les règles

Dans ce cadre, certains travaux ont eu recours à la programmation des règles linguistiques vocales, morphologiques et syntaxiques pour la voyellation des mots arabes. (El-Sadany, Hashish, 1988) ont mentionné une méthode se basant sur des règles morphologiques pour la voyellation semi-automatique des verbes arabes. Aussi, (Debili, Achour, 1998) ont étudié l'impact de l'analyse lexicale, l'analyse morphologique et l'étiquetage syntaxique pour dissiper l'ambigüité dans le processus de voyellation des textes arabes. L'absence d'un système de voyellation des textes arabes basé uniquement sur les règles est due aux taux élevé d'ambigüités, de l'existence d'un nombre important de règles morpho-syntaxiques et l'absence d'un analyseur syntaxique efficace.

2.2 Approches statistiques

(Gal 2002) a présenté une approche markovienne pour la voyellation du Coran pour la langue arabe et les textes de l'Ancien Testament pour la langue hébraïque. (Schlippe et al., 2008) ont mis au point un système de voyellation des textes arabes basé sur la traduction automatique. (Al Ghamdi et al. 2010) ont présenté le système de voyellation KAD (*The Arabic diacritizer*) basé sur les 4-grammes au niveau des lettres. Enfin (Hifny, 2013) a présenté une méthode purement statistique basée sur les n-grammes et utilisant quelques techniques de lissage qui prélèvent une masse de probabilité sur les transitions observées, et cette masse est redistribuée sur les événements non observés.

2.3 Approches hybrides

Ce sont les approches qui combinent les règles linguistiques et les traitements statistiques afin d'exploiter les points forts des deux méthodes. Parmi les travaux importants, on peut citer le système de voyellation ArabDiac développé par la société RDI (Rashwan et al., 2011). Ce système utilise l'analyseur morphologique ArabMorpho et l'étiqueteur ArabTagger puis les n-grammes. (Zitouni et al., 2009) ont présenté un système de voyellation utilisant un classificateur statistique basée sur l'entropie maximale. Leurs caractéristiques sont basées sur des caractères simples du mot, segments morphologiques et l'état syntaxique pour atteindre le meilleur classement de mots. (Habash, Rambow, 2007) utilisent la version 2.0 de l'analyseur Arabic Morphological Analyzer (Buckwalter, 2004) pour obtenir toutes les analyses morphologiques possibles. Puis ils utilisent des classificateurs individuels pour lever l'ambiguïté entre ces analyses. (Bebah et al., 2013) et (Chennoufi, Mazroui, 2014) ont présenté plusieurs approches morpho-statistiques basées sur différents états observés comme les classes des mots ou les mots non voyellés et des états cachés tels que les schèmes des mots, les listes diacritiques ou les mots voyellés. Les similitudes avec la méthode présentée dans cet article sont l'utilisation de l'analyseur morphologique AlKhalil Morpho Sys (Bebah et al., 2011) et les différences sont l'utilisation des techniques de lissage et la taille des corpus utilisés dans les phases d'apprentissage et de test.

3 Présentation du processus de voyellation automatique

Nous allons donner dans cette section une présentation détaillée du système développé. Le processus de voyellation automatique des textes de la langue arabe que nous avons surnommé AlKhalil Diacritizer se fera en trois phases principales, comme le montre la Figure 1.

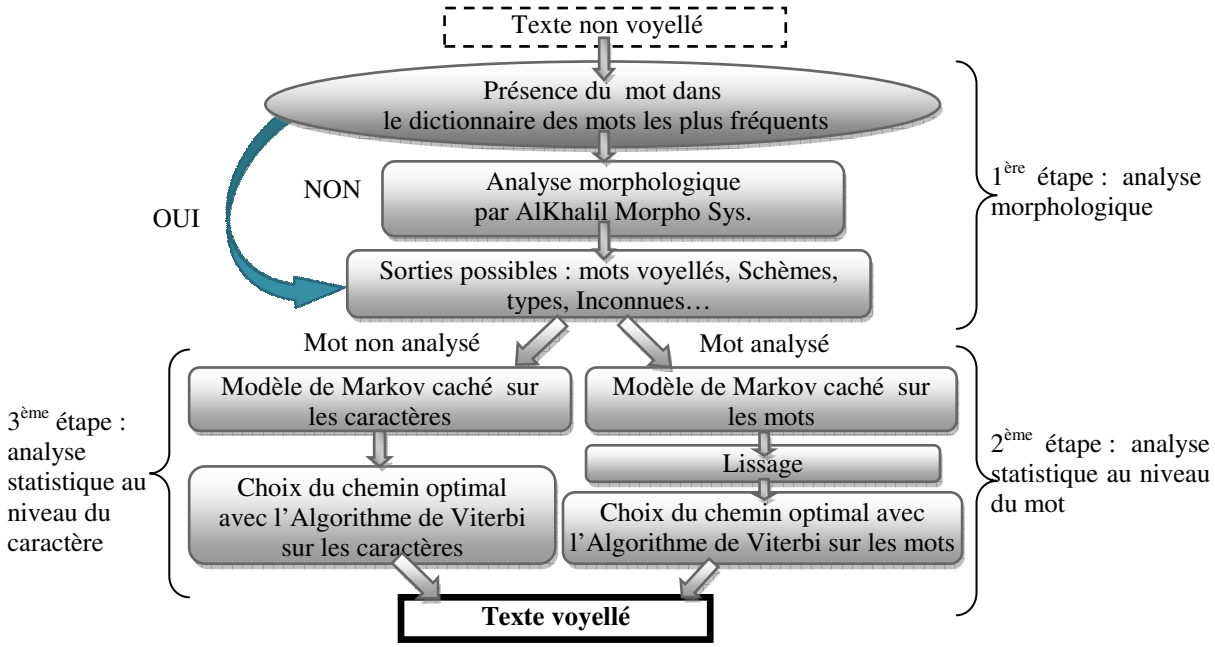


FIGURE 1 : Etapes de la voyellation automatique des textes arabes

3.1 Première phase : Analyse morphologique

L'analyse morphologique est réalisée en utilisant l'analyseur open source Alkhalil Morpho Sys[1]. Ce dernier fournit des informations morpho-syntaxiques hors contexte du mot telles que les voyellations possibles du mot, les affixes qui s'ajoutent aux stems, le stem, la nature du mot (nom, verbe ou mot outil), et dans le cas des noms et des verbes le système fournit le schème, la racine et l'état syntaxique (*POS tags*). L'intégration de l'analyseur Alkhalil Morpho Sys dans le système de voyellation automatique nous a obligés de faire des ajustements qui ont consisté principalement à l'ajout d'une base de données sous la forme d'un dictionnaire contenant les mots arabes voyellés les plus fréquents dans les corpus arabes disponibles. Cette base a été générée à partir d'une base de données composée de plus de 250 millions de mots provenant de huit corpus arabes disponibles sur Internet comprenant des livres anciens et contemporains. Le but de l'élaboration de ce dictionnaire est d'une part l'accélération du processus de l'analyse morphologique et d'autre part l'ajustement de la qualité de la voyellation des mots les plus fréquents des différents textes. Ce dictionnaire des mots les plus fréquents a atteint 16369 mots accompagnés de leurs différentes voyellations possibles (voir Figure 2).

Mot non voyellé	Les voyellations possibles du mot
ثم	ثُم ثَمَّ
المالكية	الْمَالِكِيَّةِ الْمَالِكِيَّةَ الْمَالِكِيَّةُ
يجزئ	يُجْزِئُ يُجْزِئَ يُجْزِئِ يُجَزِّئُ يُجْزِئْ

FIGURE 2 : Aperçu du dictionnaire des mots les plus fréquents

3.2 Deuxième phase : Analyse statistique au niveau du mot

Après que le programme a procédé à une analyse morphologique des mots du texte permettant d'avoir les voyellations possibles pour chaque mot, nous abordons la deuxième étape du processus de voyellation. Elle consiste à un traitement statistique se basant sur le modèle de Markov caché au niveau du mot, les techniques de lissage et l'algorithme de Viterbi (Neuhoff, 1975). Cela permet d'obtenir la voyellation la plus probable des mots de la phrase. Dans ce qui suit, nous donnons un bref rappel des démarches mathématiques relatives à ce modèle.

3.2.1 Modèle de Markov caché

Soit $O = \{o_1, \ldots, o_M\}$ un ensemble fini d'observations et soit $S = \{s_1, \ldots, s_N\}$ un ensemble fini d'états cachés.

Définition : Un modèle de Markov caché du premier ordre est un couple de processus aléatoires $(X_t, Y_t)_{t\geq 1}$ tel que $(X_t)_{t\geq 1}$ est une chaine de Markov homogène à valeurs dans l'ensemble des états cachés S, ainsi :

$$\Pr(X_{t+1} = s_j / X_t = s_i, \ldots, X_1 = s_h) = \Pr(X_{t+1} = s_j / X_t = s_i) = a_{ij} \quad (1)$$

a_{ij} = la probabilité de transition de l'état s_i à l'état s_j et la matrice $A=(a_{ij})_{ij}$ est appelée matrice de transition,
et $(Y_t)_{t\geq 1}$ est un processus observable à valeurs dans l'ensemble des observations O vérifiant :

$$\Pr(Y_t = o_k / X_t = s_i, Y_{t-1} = o_{k_{t-1}}, X_{t-1} = s_{i_{t-1}}, \ldots, Y_1 = o_{k_1}, X_1 = s_{i_1}) = \Pr(Y_t = o_k / X_t = s_i) = b_i(k) \quad (2)$$

$b_i(k)$ = la probabilité d'observer l'état o_k étant donné l'état s_i et la matrice $B=(b_i(k))_{ik}$ est appelée matrice d'émission.
Dans notre approche, les états observés du modèle markovien sont les mots arabes non voyellés et les états cachés sont les mots voyellés. Par exemple, l'état observé "حدث" peut avoir plusieurs états cachés tels que "حَدَّثَ" qui signifie "*il a raconté*" ou "حَدَثٌ" qui signifie "*événement*".

3.2.2 Méthodes de lissage

Les paramètres du modèle statistique à savoir la matrice de transition $A=(a_{ij})_{ij}$ et la matrice d'émission $B=(b_i(k))_{ik}$ seront estimés à partir de corpus linguistiques représentatifs. La méthode utilisée pour estimer les coefficients a_{ij} et $b_i(k)$ est basée sur le maximum de vraisemblance (Manning, Schütze, 1999). Ainsi, si pour un mot non voyellé u_k (état observé) et deux mots voyellés w_i et w_j (états cachés) nous notons :
$r=c(w_i,w_j)$:le nombre d'occurrences dans le corpus d'apprentissage de la transition de l'état caché w_i vers l'état caché w_j,
$c(w_i)$: le nombre d'occurrences dans le corpus d'apprentissage de l'état caché w_i,
$c(u_k,w_j)$: le nombre de fois que le mot non voyellé u_k correspond dans le corpus d'apprentissage au mot voyellé w_i,
alors les coefficients a_{ij} et $b_i(k)$ seront estimés à l'aide des équations (3) et (4) suivantes :

$$a_{ij} = \frac{c(w_i, w_j)}{c(w_i)} = \frac{r}{c(w_i)} \qquad 1 \leq i \leq N \quad 1 \leq j \leq N \quad (3)$$

$$b_i(k) = \frac{c(u_k, w_i)}{c(w_i)} \qquad 1 \leq k \leq M \quad 1 \leq i \leq N \quad (4)$$

où N (resp. M) est le nombre sans répétition des mots voyellés (resp. non voyellés) du corpus d'apprentissage.
Il faut signaler que les éléments de la matrice d'émission B sont soit égaux à 1 soit égaux à 0 car si le mot u_k dépourvu de ces voyelles est identique au mot w_i alors $b_i(k)$ est égal à *1* et dans le cas contraire il est égal à *0*. Par exemple la probabilité pour que l'état caché "فَهِمَ" qui signifie "*il a compris*" génère l'état observé qui est le mot sans voyelle "فهم" est toujours égale à *1* ($\Pr(u_k = \text{فهم} / w_i = \text{فَهِمَ}) = b_i(k) = 1$). Cet ainsi que seule les probabilités de transition de la matrice A subiront les techniques de lissage citées ci-après.

[1] www.sourceforge.net/projects/alkhalil

Durant la phase de test, nous avons constaté que certaines transitions sont non disponibles dans la matrice de transition *A*. En effet, l'estimation de toutes les transitions à partir des observations n'est pas suffisante, car il n'existe pas de corpus assez grand pour observer toutes les séquences de mots valides qui peuvent être générés par le vocabulaire. D'ailleurs, ce n'est pas parce qu'une transition est absente du corpus d'apprentissage qu'il est impossible de l'observer dans d'autres corpus. De plus, il est très contraignant, lors de la recherche de la solution optimale d'avoir une probabilité nulle pour un événement. Des techniques de lissage sont alors utilisées pour combler ces lacunes et permettre au modèle de langage d'attribuer une probabilité non nulle à toutes les transitions. Ces techniques sont appliquées avant de faire tourner l'algorithme de Viterbi car sinon le modèle est moins performant lorsque l'une des probabilités est nulle. L'algorithme de Viterbi permet d'identifier le chemin optimal dans le réseau des solutions parmi les voyellations possibles du mot (voir Figure 3).

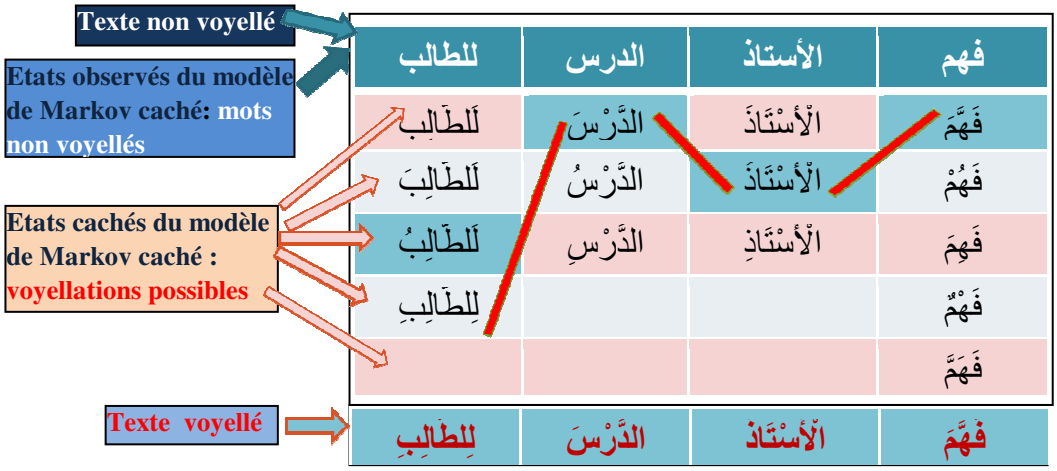


FIGURE 3. Utilisation de l'algorithme de viterbi pour trouver la solution optimale

La plupart des méthodes de lissage peuvent se décomposer en deux étapes : la première prélève une masse de probabilité sur les transitions observées (probabilité réduite), et cette masse est redistribuée dans la seconde étape sur les événements non observés. Pour le prélèvement, beaucoup de techniques ont été expérimentées (Chen, Goodman, 1998). La redistribution se fait généralement soit par interpolation linéaire soit par repli à l'ordre inférieur (plus connue sous l'appellation back-off). Dans cet article, nous allons nous limiter à la forme interpolée. Ainsi, l'équation (5) de lissage pour les probabilités de transition du modèle de Markov caché d'ordre 1 est la suivante:

$$P_{\text{int erpolation}}(w_j / w_i) = \tau(w_j|w_i) + \alpha(w_i) P_{\text{int erpolation}}(w_j) \qquad (5)$$

$P_{\text{int erpolation}}(w_j) = P_{MLE}(w_j)$ ou $P_{\text{int erpolation}}(w_j) = P_{uniforme}(w_j) = \frac{1}{|V|}$ où le vocabulaire V est l'ensemble de tous les mots.

$P_{MLE}(w_j)$ = probabilité du mot w_j calculée par la méthode du maximum de vraisemblance (Manning, Schütze,, 1999).

$\tau(w_j|w_i)$ = la probabilité réduite des transitions observées et $\alpha(w_i)$ est le poids de repli (*backoff weight*).

3.2.2.1 Méthode Additive Smoothing

Le lissage le plus simple utilisé dans la pratique est l'Additive Smoothing (Chen, Goodman, 1998). Pour éviter les transitions nulles, nous faisons l'hypothèse que chaque transition se produit un peu plus souvent qu'il ne parait. Nous ajoutons un facteur δ pour chaque fréquence de transition, où généralement $0 < \delta \leq 1$. Ainsi, l'équation est la suivante :

$$P_{add}(w_j / w_i) = \frac{\delta + c(w_i w_j)}{\delta|V| + c(w_i)} \qquad (6)$$

La performance de cette méthode de lissage est faible, car celle-ci a tendance à surestimer les probabilités des événements absents dans le corpus d'apprentissage (Gale, Church, 1994).

3.2.2.2 Lissage Absolute discounting

La méthode Absolute discounting fait partie des méthodes de lissage interpolé (Ney et al., 1994) qui est obtenu en prélevant une constante D comprise entre 0 et 1 de chaque probabilité non nulle. La combinaison de la probabilité d'ordre supérieur et d'ordre inférieur pour les transitions des chaines de Markov cachées d'ordre 1 est donnée par :

$$P_{abs}(w_j / w_i) = \frac{\max\{r - D, 0\}}{c(w_i)} + \frac{D}{c(w_i)} N_{1+}(w_i \bullet) P_{abs}(w_j) \qquad 0 \leq D \leq 1$$

Avec selon les auteurs : $P_{abs}(w_j) = P_{MLE}(w_j)$ ou $P_{abs}(w_j) = P_{uniforme}(w_j) = \frac{1}{|V|}$,

et $N_{1+}(w_i \bullet)$ est la diversité qui est définie comme étant le nombre de tous les mots (sans répétition) qui suivent le mot w_i dans le corpus.

$$N_{1+}(w_i \bullet) = \left|\{w_j : c(w_i w_j) > 0\}\right| \qquad (7)$$

3.3 Troisième phase : Analyse statistique au niveau du caractère

Durant la phase de test, nous avons rencontré une autre contrainte liée aux mots non analysés par Alkhalil Morpho Sys et pour lesquels nous avons associé l'étiquette "*unkown*". De ce fait, la troisième phase d'AlKhalil Diacritizer ne concerne que ces cas. Ces derniers ne sont pas voyellés par la 2ème phase de notre système. Ainsi, pour chaque mot non analysé, nous utilisons un autre modèle de Markov caché dont les observations sont les lettres arabes et les états cachés sont les signes diacritiques. L'algorithme de Viterbi est utilisé également pour le choix de la solution optimale.

4 Etapes d'apprentissage et de test du modèle

La phase d'apprentissage a été réalisée sur 90% d'un corpus formé de plus de 63 millions de mots voyellés du corpus Tashkeela[2] (56 939 523 mots). Les 10% restants (6 306 237 mots) seront utilisés dans la phase de test. Ce corpus est composé de textes voyellés tirés d'anciens livres traitant des sujets comme la théologie, la grammaire, l'histoire, l'économie et la géographie. Ensuite, nous avons réalisé certaines opérations qui consistent à segmenter ces textes en phrases puis en déduire les statistiques sur les voyellations des mots pour l'estimation des paramètres du modèle.

Afin d'évaluer les performances globales de notre voyelliseur , nous avons testé les méthodes de lissage sur un corpus test dépassant six millions de mots (6 306 237 mots) et choisit aléatoirement du corpus d'apprentissage Tashkeela. Les résultats (voir table 1) montrent que lorsque le lissage n'est pas utilisé, le taux d'erreurs au niveau des mots WER1 (WER: Word Error Rate) est de 26.98% et baisse pour atteindre 14.02% (WER2) lorsque nous ignorons le signe diacritique de la dernière lettre du mot. De même, le taux d'erreurs DER1 (DER: Diacritic Error Rate) relatif à tous les caractères du texte est de l'ordre de 10.19% et celui qui ne tiens pas compte du signe diacritique du dernier caractère (DER2) est de l'ordre de 5.46%. Après application de la méthode Additive Smoothing, les quatre taux d'erreurs diminuent et sont de l'ordre de 12% pour WER1, de 7.5% pour WER2, de 4.7% pour DER1 et 2.9% pour DER2. Il reste à signaler que cette méthode donne les meilleurs scores pour δ=0.1. En utilisant la forme interpolée de la méthode Absolute Discounting et $P_{MLE}(w_j)$ comme probabilité de repli à l'ordre inférieur, nous avons amélioré de deux points les performances du système. Ainsi, les quatre taux d'erreurs obtenus sont de l'ordre 10% pour WER1, de 5.4% pour WER2, de 3.72% pour DER1 et de 2.09% pour DER2. Ces performances sont obtenues avec le paramètre D=0.5.

Méthode de lissage	WER1 (%)	WER2(%)	DER1(%)	DER2(%)
Sans lissage	26.98	14.02	10.19	5.46
Additive Smoothing (δ=0.1)	12.06	7.37	4.73	2.87
Additive Smoothing (δ=0.5)	12.35	7.45	4.82	2.90
Additive Smoothing (δ=1)	12.65	7.56	4.92	2.94
Absolute Discounting (D=0.1)	10.11	5.43	3.73	2.09
Absolute Discounting (D=0.5)	**10.06**	**5.40**	**3.72**	**2.09**
Absolute Discounting (D=1)	11.16	5.76	4.09	2.22

TABLE 1 : Résultats de l'évaluation d'AlKhalil Diacritizer avec les différentes techniques de lissage

Il ressort de ces résultats que l'utilisation des techniques de lissage améliore considérablement les performances du système de vocalisation automatique des textes arabes AlKhalil Diacritizer. La méthode Absolute Discounting affiche les taux d'erreurs les plus faibles comparativement à la méthode Additive Smoothing. D'autre part, pour comparer nos résultats à ceux des autres systèmes de la littérature, nous donnons dans la table 2 les différents taux d'erreurs. Cependant, vu que ces systèmes n'ont pas été testés sur le même corpus, il conviendra de prendre les conclusions avec une certaine réserve.

Système de voyellation	WER1 (%)	WER2(%)	DER1(%)	DER2(%)
(Schlippe et al., 2008)	13.80	9.30	4.90	3.20
(Zitouni et al., 2009)	17.30	7.20	5.10	2.20
(Al Ghamdi et al. 2010)	46.83	26.03	13.83	9.25
(Rashwan et al., 2011)	12.50	**3.10**	3.80	**1.20**
Notre système AlKhalil Diacritizer	**10.06**	5.40	**3.72**	2.09

TABLE 2 : Comparaison des performances d'AlKhalil Diacritizer avec certains voyelliseurs de la littérature

Enfin, nous pouvons avancer certaines remarques expliquant les taux d'erreur de notre modèle. La mesure WER1 telle que nous l'avons adoptée pour évaluer le système est une mesure très exigeante puisque la voyellation du mot est considérée correcte s'il y a concordance totale entre le mot d'origine et le mot résultat du système de voyellation. Or, les corpus disponibles présentent plusieurs erreurs orthographiques ayant comme conséquence un impact négatif sur les performances de l'analyseur morphologique intégré. D'autre part, la diminution du taux WER2 montre que presque la moitié des erreurs de voyellation (5.40% sur 10.06%) sont des erreurs syntaxiques (erreur relative au dernier caractère).

[2] http://sourceforge.net/projects/tashkeela/

5 Conclusion et perspectives

Nous avons présenté dans ce papier un programme de voyellation automatique basé sur une approche hybride qui combine l'analyse morphologique et les modèles de Markov cachés. Le modèle a été appris sur un corpus représentatif constitué de livres arabes d'environ 57 millions de mots voyellés. Ensuite plusieurs techniques de lissage ont été appliquées sur les paramètres de ce modèle pour contourner le problème des transitions de mots non vues dans le corpus. Les résultats d'évaluations obtenus sont très encourageants en comparaison avec d'autres systèmes disponibles. Nous prévoyons de les améliorer en agissant sur plusieurs niveaux :

- Au niveau des corpus : d'une part nous allons les enrichir par d'autres textes, et d'autre part nous essayerons de corriger les erreurs orthographiques.
- Au niveau du dictionnaire des mots les plus fréquents : nous essayerons de compléter les voyellations de certains mots partiellement voyellés.
- Au niveau de l'analyse linguistique : nous chercherons à réduire le taux d'erreurs relatif au signe diacritique du dernier caractère du mot en exploitant des informations syntaxiques données par l'analyseur Alkhalil Morpho Sys.

Références

Alghamdi M., Muzaffar Z., Alhakami H. (2010), "Automatic Restoration of Arabic Diacritics: A Simple, Purely Statistical Approach," The Arabian Journal for Science and Engineering, vol. 35, 2010.

Buckwalter T. (2004). Arabic Morphological Analyzer version 2.0. LDC2004L02.

Chen S. F., Goodman J. (1998). An empirical study of smoothing techniques for language modeling, Harvard Univ.,Computer Science Group, Cambridge, MA, Tech. Rep. TR-10-98, August 1998.

Chennoufi A., Mazroui A. (2014). Vers un Voyaliseur Automatique des Textes Classiques de la Langue Arabe. 1ère Journée Doctorale Nationale sur L'ingénierie de la Langue Arabe, (JDILA'14), 8 Février 2014, Rabat, Maroc.

Debili F., Achour H. (1998). Voyellation automatique de l'arabe. In Proceedings of the workshop on Computation approaches to Semitic languages, COLING-ACL '98. Montréal.

El-Sadany T., Hashish M. (1988). Semi-automatic vowelization of arabic verbs. In 10th NC Conference, Saudi Arabia.

Gale W., Church K.(1994). What is wrong with adding one? In Corpus-Based Research into Language, N. Oostdijk and P. D. Haan, Eds. Amsterdam, The Netherlands: Rodopi, 1994, pp. 189–198.

Gal Y. (2002). An HMM Approach to Vowel Restoration in Arabic and Hebrew. In ACL-02 Workshop on Computational Approaches to Semitic Languages.

Habash N., Rambow O.(2007), "Arabic diacritization through full morphological tagging," in Proceedings of NAACL/HLT 2007. Companion Volume, Short Papers, Rochester, New York, USA. April. 2007.pages53-56

Hifny Y.(2013). Restoration of arabic diacritics using dynamic programming. In The 8th International Conference on Computer Engineering & Systems (ICCES'2013), 26-28 Nov. 2013, Cairo, Egypt, pages 3-8 ISBN:978-1-4799-0078-7.

Maamouri M., Bies A., Kulick S. (2006). Diacritization: A Challenge to Arabic Treebank Annotation and Parsing. In Proceedings of the British Computer Society Arabic NLP/MT Conference.

Manning C. , Schutze H. (1999). Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA.

Messaoudi A., Lamel L., Gauvain J. (2004). The LIMSI RT04 BN Arabic system. In Proc. Darpa RT04, Palisades NY.

Neuhoff D.L. (1975). The Viterbi Algorithm as an Aid in Text Recognition. In IEEE Transaction on InformationTheory. Pages 222-226.

Ney H., Essen U., Kneser R. (1994). On structuring probabilistic dependences in stochastic language modeling. Computer, Speech, and Language, vol. 8, pp. 1{38, 1994.

Ould Bebah M. O. A, Mazroui. A, Meziane A., Lakhouaja A. (2011). Alkhali Morpho Sys. In International Computing Conference in Arabic. Riadh, Arabie Saoudite.

Ould Bebah M. O. A., Mazroui A ., Lakhouaja A., Chennoufi A.(2013). Approche morpho-statistique pour la voyellation automatique des texts arabes. In 8th International Conference on intelligent system: theories and applications (SITA'13), May 08-09, 2013, EMI, Rabat , Morocco.

Vergyri D., Kirchoff K. (2004). Automatic diacritization of Arabic for Acoustic Modeling in Speech Recognition. In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. Geneva. Pages 66-73.

Rashwan M., Al-Badrashiny M. , Attia M., Abdou S.M., Rafea A. (2011), "A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features," Audio, Speech, and Language Processing, pages. 166-175.

Schlippe T., Nguyen T. , Vogel S. (2008). Diacritization as a Machine Translation Problem and as a Sequence Labeling Problem. In 8th AMTA conference, Hawaii. Pages 21-25.

Zitouni I., Sarikaya R. (2009), "Arabic Diacritic Restoration Approach Based on Maximum Entropy Models," Computer Speech & Language, vol. 23, no. 3, pp. 257-276, 2009.

Évaluation d'un système d'extraction de réponses multiples sur le Web par comparaison à des humains

Mathieu-Henri Falco Véronique Moriceau Anne Vilnat
LIMSI-CNRS, Université Paris-Sud, 91405 Orsay, France
prenom.nom@limsi.fr

Résumé. Dans cet article, nous proposons une évaluation dans un cadre utilisateur de Citron, un système de question-réponse en français capable d'extraire des réponses à des questions à réponses multiples (questions possédant plusieurs réponses correctes différentes) en domaine ouvert à partir de documents provenant du Web. Nous présentons ici le protocole expérimental et les résultats pour nos deux expériences utilisateurs qui visent à (1) comparer les performances de Citron par rapport à celles d'un être humain pour la tâche d'extraction de réponses multiples et (2) connaître la satisfaction d'un utilisateur devant différents formats de présentation de réponses.

Abstract. In this paper, we propose a user evaluation of Citron, a question-answering system in French which extracts answers for multiple answer questions (expecting different correct answers) in open domain from Web documents. We present here our experimental protocol and results for user evaluations which aim at (1) comparing multiple answer extraction performances of Citron and users, and (2) knowing user preferences about multiple answer presentation.

Mots-clés : système de question-réponse, réponses multiples, évaluation utilisateur.

Keywords: question-answering system, multiple answers, user evaluation.

1 Introduction

Nous nous intéressons à l'évaluation de Citron, un système de question-réponse (SQR) en français capable de répondre à des questions à réponses multiples (*question-ARM*) en domaine ouvert à partir de documents provenant du Web. Les questions-ARM sont des questions possédant plusieurs réponses correctes différentes (par exemple *Quand le PSG a-t-il remporté la Coupe de France de football ?*), les questions de type liste (*question-liste*) en sont un exemple (*Quelles sont les planètes du système solaire ?*). Nous avons constaté en étudiant les données des campagnes d'évaluations des SQR pour le français comportant des questions-listes (Ayache, 2005), (Quintard *et al.*, 2010) que toutes les réponses à une question-liste se trouvaient quasiment toujours à l'intérieur d'un même document et étaient même concentrées majoritairement dans une seule phrase (Falco, 2012). Ce constat va de pair avec l'évaluation faite par ces campagnes qui impose un seul support justificatif par réponse, limité en caractères et continu, rendant alors impossible des recoupements multi-documents pour justifier une réponse. Ces contraintes sont d'autant plus pénalisantes lorsque les documents utilisés pour extraire les réponses sont issus du Web. En effet, ces documents sont des sources non négligeables de réponses aux questions-ARM grâce aux nombreuses structures (énumérations, tableaux) qu'ils contiennent. Pourtant les formats de réponse imposés par les campagnes d'évaluation ne permettent pas d'exploiter ces structures entièrement.

Citron exploite les structures (énumérations, tableaux) propres à ce type de documents pour mieux extraire les réponses et éventuellement les agréger pour faire apparaître des critères variants comme la date ou le lieu (Falco, 2012). Nous avons déjà évalué notre système selon les critères des campagnes d'évaluation (précision, rappel) et les résultats obtenus sont encourageants (Falco *et al.*, 2013). Pour la tâche d'extraction de réponses à des questions-ARM, nous souhaitons comparer les performances de Citron à celles d'un humain. L'hypothèse intuitive est qu'un système automatique est plus rapide mais peut-être un peu moins bon qu'un humain pour extraire les réponses. Nous avons surtout souhaité évaluer un aspect original de Citron qui n'est pas pris en compte durant les campagnes d'évaluation à savoir la présentation des réponses extraites. En effet, lorsque plusieurs réponses sont correctes, nous pensons qu'apporter le critère variant expliquant pourquoi il existe plusieurs réponses correctes est important pour la compréhension d'un utilisateur. Par une deuxième expérience, nous avons donc souhaité connaître la satisfaction des utilisateurs concernant le format de présentation des ré-

ponses imposé par les campagnes et celui de Citron. Dans cet article, nous présentons donc rapidement Citron et détaillons les protocoles de nos deux expériences en cadre utilisateur. Enfin, nous discutons leurs résultats.

2 Le système de question-réponse Citron

Citron est un SQR spécialisé dans l'extraction de réponses à des questions-ARM en français. Il peut travailler sur des collections de documents de différents types ; nous nous intéressons ici à une collection de documents issus du Web prétraités par le programme Kitten (Falco *et al.*, 2012) qui permet notamment de repérer les éléments structuraux susceptibles de contenir des réponses multiples (structures énumératives (SE) et tableaux) des documents Web et de les formater de manière exploitable pour une analyse syntaxique. Citron utilise l'analyseur XIP (Aït-Mokhtar *et al.*, 2002) qui permet d'obtenir une analyse en dépendances et intègre également un détecteur d'entités nommées. L'analyse de la question avec XIP vise à extraire les termes importants de la question et permet de générer une requête soumise à Lucene (Hatcher *et al.*, 2010) pour trouver des snippets susceptibles de contenir la réponse. Cette recherche se fait dans 3 index différents : l'index des parties "texte" des documents, l'index des SE et celui des tableaux. XIP analyse ensuite les snippets en dépendances. Citron utilise les règles syntaxiques et lexiques de XIP définis pour le SQR FIDJI (Moriceau & Tannier, 2010) développé précédemment que nous avons complétés notamment pour analyser les éléments structuraux prétraités par Kitten. La validation du type des candidats-réponses utilise Wikipédia dans une approche similaire à (Grappy, 2011) mais nous n'utilisons que l'introduction de l'article correspondant au candidat à valider pour y rechercher des définitions. Finalement, l'agrégation de réponses a pour but de regrouper des réponses désignant une même entité. Citron utilise une approche surfacique et une normalisation des dates contenues dans les supports des réponses effectuée à l'aide de HeidelTime (Strötgen & Gertz, 2013). Cette agrégation permet également de regrouper les réponses selon un critère variant temporel (Falco, 2012) (voir partie gauche de la figure 1). Nous ne détaillons pas plus ici le fonctionnement de Citron qui est décrit précisément dans (Falco *et al.*, 2013).

3 Expériences utilisateur : présentation du protocole

Nous présentons dans cette section le protocole expérimental utilisé pour nos deux expériences utilisateurs réalisées consécutivement : (1) l'extraction de réponses (*extraction*) : son objectif est de comparer les performances de Citron par rapport à celles d'un être humain pour la tâche d'extraction de réponses multiples depuis une collection de documents imposés issus du Web. Cette première expérience permet surtout d'étudier ce que les utilisateurs considèrent comme des réponses correctes et comment ils les rédigent/présentent ; et (2) la satisfaction devant la présentation de réponses (*satisfaction*) : son objectif est de connaître la satisfaction d'un utilisateur devant des réponses extraites et formatées de deux façons différentes, l'une au format d'une campagne d'évaluation et l'autre telle que Citron les présenterait.

3.1 Données d'évaluation et outils

Nous avons généré deux jeux (*jeuA* et *jeuB*) de 10 questions-ARM homogènes au niveau des types de questions proposées : 5 questions dont le type attendu de la réponse est général (une entité nommée) et 5 dont le type est spécifique. Chaque question-ARM d'un jeu de questions a son miroir syntaxique dans l'autre jeu avec un changement de focus :
– type attendu général : *Qui a joué Knocking on Heaven's Door ?* (A), *Qui a joué Where Did you Sleep last Night ?* (B)
– type attendu spécifique : *Quels sont les **signes** du zodiaque ?* (A), *Quels sont les **péchés** capitaux ?* (B)
Pour chaque utilisateur, un jeu sert pour l'expérience *extraction* et l'autre pour *satisfaction*. Pour chaque question, nous avons indexé grâce à Lucene les 30 premiers documents renvoyés par Google pour la requête générée par Citron et prétraités par Kitten. À l'aide de cette même requête sur les 3 index (texte, SE et tableaux), on obtient une liste de snippets pour chaque question. Nous avons ensuite utilisé WebAnnotator (Tannier, 2012) afin d'annoter toutes les occurrences de réponses jugées correctes, soit au total 1084 occurrences. Puis nous les avons agrégées lorsque nécessaire (par exemple, *Olympique de Marseille* et *OM* désignent une même entité). Ces 1084 réponses correctes correspondent à 280 entités différentes.

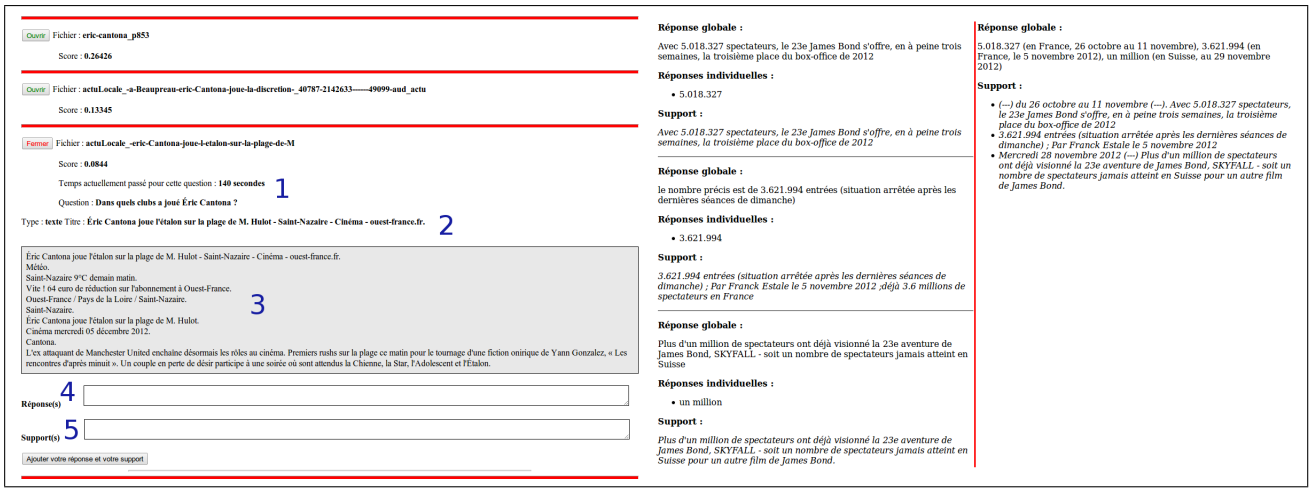


FIGURE 1 – À gauche : zone de saisie de réponses pour l'expérience d'*extraction*. À droite : présentation des réponses en deux colonnes pour l'expérience *satisfaction* sur la question *Combien de spectateurs ont vu Skyfall ?* (Formatage campagne d'évaluation à gauche, formatage Citron à droite).

3.2 Interface d'utilisation

Les deux expériences d'*extraction* et de *satisfaction* se font par l'intermédiaire d'une interface Web que nous avons développée avec le framework Django [1]. Un tutoriel était proposé avant le début de chaque expérience. La figure 1 présente à gauche l'interface pour l'évaluation de l'extraction de réponses et à droite celle pour la présentation des réponses.

Dans l'interface pour l'extraction des réponses, pour chaque question, les documents sont présentés les uns en dessous des autres et fermés au début. L'utilisateur choisit les documents à ouvrir et une fois un document ouvert, s'affichent le temps écoulé depuis le début de la question et le type du snippet (texte, tableau ou SE). Si l'utilisateur pense que le snippet contient une réponse correcte, il peut l'extraire avec un support (défini par *ce que vous pensez être le texte justifiant la ou les réponses extraites*) dans la zone de formulaire. Si un snippet contient plusieurs réponses correctes, l'utilisateur peut saisir une réponse à la fois ou plusieurs réponses en même temps. Les contraintes sont que la réponse doit être extraite depuis le snippet et qu'elle doit être accompagnée d'un support non vide. L'utilisateur est informé qu'un snippet ne contient pas forcément de réponse, qu'il y a au moins deux réponses/entités différentes pour chaque question et qu'il peut passer à la question suivante dès qu'il le souhaite, y compris sans fournir la moindre réponse.

Les données présentées dans la seconde interface sont les réponses de deux utilisateurs durant l'expérience *extraction* formatées de deux façons : une selon les critères de la campagne Quaero 2009 et l'autre tel que Citron les présenterait :
– format Quaero : de 1 à 3 bloc-réponses. Un bloc-réponse se compose d'une réponse globale continue (250 caractère maximum), d'un support continu (8 000 caractères maximum) et de n réponses individuelles provenant du support ;
– format Citron : une réponse globale composée des n réponses individuelles choisies par les deux utilisateurs et une liste de supports (pas forcément continus). Chaque réponse individuelle peut être accompagnée de critères variants.

Chaque formatage est présenté aléatoirement cinq fois à droite et cinq fois à gauche. Des questions sont posées à l'utilisateur pour lui demander quel format de réponse il préfère.

3.3 Les utilisateurs

32 utilisateurs ont passé les 2 expériences : 46,88 % ont passé l'expérience *extraction* sur le *jeuA* et *satisfaction* sur le *jeuB*. Avant de commencer l'expérience, l'utilisateur répond à 6 questions dans le but de définir son profil : sa tranche d'âge, s'il travaille dans la recherche, dans le TAL, combien d'articles scientifiques portant sur les SQR il a lus [2], combien de requêtes il effectue sur un moteur de recherche quotidiennement et son niveau d'informatique [3].

1. `http://djangoproject.com`
2. Nous avons considéré qu'une personne était sensibilisée à la thématique question-réponse si elle a lu au moins un article.
3. Le niveau *normal* correspond à une utilisation quotidienne d'Internet et d'un logiciel de bureautique, celui *intermédiaire* implique l'utilisation d'un programme plus poussé (retouche d'image, musique).

10-25 ans	26-40 ans	41 ans et +	Dans la recherche	Dans le TAL	Lecture SQR	+ de 10 requêtes	niveau normal	niveau intermédiaire	niveau avancé
28,13 %	53,12 %	18,75 %	59,38 %	37,50 %	53,13 %	51,61 %	31,25 %	12,50 %	56,25 %

TABLE 1 – Répartition des profils utilisateurs.

4 Résultats de l'extraction des réponses

4.1 Annotations des réponses et mesures d'évaluation

Les utilisateurs étaient libres de saisir dans le formulaire une seule réponse à la fois ou plusieurs en même temps tant qu'elles provenaient du même snippet. Nous avons donc d'abord segmenté toutes les réponses fournies en réponses individuelles et sommes arrivés à un total de 2319 réponses/entités individuelles, soit une moyenne de 72,47 réponses individuelles par utilisateur et de 7,27 réponses par question. Ces 2319 réponses proviennent pour 33,72 % de structures énumératives, 24,28 % de tableaux et 42 % de texte. Nous les avons annotées manuellement avec les statuts de réponses de la campagne Quaero que nous avons complétés ainsi (le statut Quaero est donné entre parenthèses) :

– (full) **Correct full** : la réponse est correcte, le support valide entièrement la réponse ;
– (full) **Correct OK** : la réponse est correcte, le support valide presque la réponse. Par exemple, pour *Quelles villes ont été la capitale des États-Unis ?*, la réponse *Philadelphie* associée au support *Philadelphie, capitale originelle* ne permet pas de savoir qu'il est question des États-Unis sans regarder le snippet ;
– (right) **Correct** : la réponse est correcte, provient du snippet mais le support ne valide pas la réponse ;
– (unsupported) **Correct absente extrait** : la réponse est correcte mais n'a pas été extraite depuis le snippet ;
– (inexact) **Correct non segmentée** : la réponse est correcte mais mal segmentée. C'est le cas par exemple quand on fournit une phrase complète sans extraire ce qui est considérée comme la réponse ;
– (inexact) **Correct pb orthographe** : la réponse est correcte mais contient une faute d'orthographe ;
– (inexact) **Correct rédigée** : la réponse est correcte mais l'utilisateur l'a rédigée au lieu de simplement l'extraire. Par exemple : *C'est NIRVANA qui a joué "Where did you sleep last night"* ;
– (supported) **Incorrect mais support correct** : la réponse est incorrecte mais le support contient une réponse correcte ;
– (false) **Incorrect contradiction** : la réponse est incorrecte et il était possible de s'en rendre compte par recoupement avec un autre extrait proposé (aucune occurrence dans cette évaluation) ;
– (false) **Incorrect** : la réponse est incorrecte.

La campagne Quaero ne comptait comme réponse valide que les réponses obtenant le statut *right* ou *full*. De notre côté, nous avons réalisé deux évaluations : une évaluation *QR* où seules les réponses ayant les statuts *Correct full*, *Correct OK* et *Correct* sont considérées comme valides (nous mesurons ainsi la tâche réelle d'extraction correcte de réponse ainsi que sa justification par le support) et une évaluation *humaine* où toute réponse dont le statut commence par *Correct* est considérée comme valide (ce sont les réponses qui peuvent être jugées acceptables par des utilisateurs).

Le tableau 2 présente les statuts des réponses extraites par Citron et par les 32 utilisateurs. Nous y constatons que par une évaluation *humaine*, il n'y a quasiment pas de réponses incorrectes parmi celles extraites par les utilisateurs (3,11 %). On constate également que quelques réponses ont été rédigées, principalement pour apporter des précisions, notamment en terme de critère variant. Par exemple pour le nombre de spectateurs ayant vu Skyfall, des utilisateurs ont ajouté la date ou le lieu comme Citron le fait (figure 1). Ce comportement est le résultat le plus important de cette évaluation puisque pour 65 % des réponses rédigées (11 sur 17), les utilisateurs ont ajouté naturellement un critère variant tout comme le propose Citron. Enfin, du point de vue de l'évaluation *QR*, les utilisateurs ont extrait 86,11 % de réponses valides uniquement avec leur support, ce qui est au-dessus des performances de Citron (72,34 %). Contrairement aux utilisateurs, Citron fournit une proportion importante de réponses incorrecte (27,66 %), les causes de ces erreurs sont détaillées juste après.

Nous avons utilisé la F-mesure pour évaluer la qualité de l'extraction des réponses. Nous avons aussi souhaité comparer les performances de Citron en terme de vitesse d'exécution par rapport à celles humaines. Pour cela nous avons mesuré les vitesses de Citron et ceux des utilisateurs pour répondre à chaque question. Pour chaque utilisateur, nous comparons sa vitesse par rapport au plus rapide des utilisateurs à fournir une réponse correcte : $V = min(\frac{n_{rapide}}{n_{utilisateur}}, 1)$ où, pour une question, n_{rapide} est le nombre de secondes le plus petit nécessaire pour fournir une réponse correcte parmi tous les utilisateurs humains et $n_{utilisateur}$ le nombre de secondes passées par l'utilisateur (le minimum sert pour Citron qui est très souvent plus rapide que le plus rapide des utilisateurs).

	Moyenne des deux jeux (% (#))		***JeuA*** (% (#))		***JeuB*** (% (#))	
Source / **Statut**	Citron	Utilisateurs	Citron	Utilisateurs	Citron	Utilisateurs
Correct full	72,36 (17)	46,74 (1084)	72 (18)	53,77 (556)	72,72 (16)	41,09 (528)
Correct OK	0	22,42 (520)	0	12,00 (124)	0	30,82 (396)
Correct	0	16,95 (393)	0	25,53 (264)	0	10,04 (129)
Correct absente extrait	0	0,26 (6)	0	0,19 (2)	0	0,31 (4)
Correct non segmentée	0	7,98 (185)	0	4,93 (51)	0	10,42 (134)
Correct pb orthographe	0	0,69 (16)	0	0,29 (3)	0	1,01 (13)
Correct rédigée	0	1,85 (43)	0	0,10 (1)	0	3,27 (42)
Incorrect mais support correct	16,82 (4)	1,34 (31)	20 (5)	0,58 (6)	13,64 (3)	1,95 (25)
Incorrect	10,82 (2,5)	1,77 (41)	8 (2)	2.61 (27)	13,64 (3)	1,09 (14)

TABLE 2 – Répartition des réponses individuelles des 32 utilisateurs et de Citron.

4.2 Résultats

Profil	$\mathbf{P_{hum}}$	$\mathbf{R_{hum}}$	$\mathbf{F_{hum}}$	$\mathbf{P_{QR}}$	$\mathbf{R_{QR}}$	$\mathbf{F_{QR}}$	$\mathbf{V_{QR}}$
TAL	0,91	0,61	0,68	0,87	0,59	**0,65**	**0,22**
Non TAL	0,86	0,55	0,62	0,77	0,51	0,57	0,21
Recherche	0,94	0,67	0,73	0,87	0,64	**0,69**	0,19
Non Recherche	0,81	0,45	0,52	0,71	0,42	0,48	**0,22**
Nb requête quotidienne <=10	0,83	0,47	0,54	0,73	0,43	0,49	**0,21**
Nb requête quotidienne > 10	0,93	0,67	0,73	0,88	0,64	**0,70**	0,20
Niveau informatique normal	0,80	0,47	0,53	0,76	0,45	0,52	**0,24**
Niveau informatique intermédiaire	0,86	0,38	0,47	0,68	0,33	0,40	0,21
Niveau informatique avancé	0,95	0,68	0,75	0,87	0,65	**0,70**	0,18
Âge 10-25 ans	0,89	0,59	0,65	0,76	0,54	0,59	**0,22**
Âge 26-40 ans	0,93	0,64	0,71	0,88	0,62	**0,68**	0,21
Âge 41 ans et plus	0,76	0,42	0,47	0,69	0,40	0,45	0,16
Aucun article QR lu	0,81	0,51	0,57	0,75	0,49	0,54	**0,23**
De 1 à 5 articles QR lus	0,91	0,63	0,69	0,82	0,59	0,65	0,20
>= 6 articles QR lus	0,95	0,66	0,73	0,92	0,65	**0,72**	0,19
Moyenne humaine (sur la totalité des questions)	0,88	0,58	0,65	0,81	0,55	**0,61**	0,21
Moyenne Citron (sur la totalité des questions)	0,47	0,36	0,37	0,47	0,36	0,37	**0,89**
Moyenne humaine (sur les questions auxquelles Citron répond)	0,91	0,64	0,70	0,85	0,61	**0,66**	0,17
Moyenne Citron (sur les questions auxquelles Citron répond)	0,72	0,56	0,56	0,72	0,56	0,56	**0,88**

TABLE 3 – Moyennes de *jeuA* et *jeuB* (*hum* pour l'évaluation des réponses sur des critères humains et *QR* selon les critères d'une campagne d'évaluation). P pour précision, R pour rappel, F pour F-mesure, V pour vitesse

.

Le tableau 3 présente les résultats pour les deux jeux de question. La **rapidité** se lit à l'aide de la mesure V_{QR} et montre que plus les utilisateurs sont jeunes, plus ils ont été rapides (+31 et +38 % pour les 2 tranches d'âge les plus jeunes). Il faut noter que la quasi-totalité des utilisateurs a choisi d'extraire le maximum de réponses différentes possibles pour chaque question avec une durée moyenne de 238,37 secondes pour une médiane de 239,77. Par exemple, seul 9 % des utilisateurs n'ont pas répondu à une question de leur jeu. Citron a été conçu pour répondre de façon rapide aux questions et se retrouve sans surprise le plus rapide, Citron passant en moyenne 22 secondes [4] par question. Même pour les questions auxquelles Citron décide de ne pas répondre, le temps consommé pour extraire les candidats-réponses et finalement ne pas les proposer ne le désavantage pas au niveau temporel. Le coefficient de corrélation de Bravais-Pearson montre qu'il n'y a pas de corrélation entre le temps passé et la qualité de l'extraction.

Pour la **qualité de l'extraction des réponses**, on constate que la F-mesure F est naturellement plus élevée selon les critères d'évaluation *humain* plutôt que *QR*. Citron est nettement moins performant principalement parce qu'il ne répond

4. 16 processeurs INTEL QUAD 5570, 50 Go de RAM, Ubuntu SMP

pas à 7 questions sur les 20. Pour ces questions, Citron a notamment rencontré des problèmes d'absence de candidats-réponses du type attendu et de non application de nos règles XIP (problème de conversion HTML, question en anglais). Citron a une forte précision : si on ne prend en compte que les questions auxquelles Citron a répondu, il n'est alors en moyenne que de 10 points en F-mesure derrière les utilisateurs, obtenant même un meilleur score pour le *jeuB*.

5 Résultats de la satisfaction des utilisateurs

Le tableau 4 montre que 71,39 % des utilisateurs ont trouvé que la présentation des réponses *à la* Citron répondait mieux à la question. La présentation *à la* Citron est vue comme justifiant le mieux ses réponses à 61,80 %, ce qui semble confirmer l'intérêt d'ajouter le critère variant aux réponses extraites quand il existe.

En ne prenant en compte que les réponses globales, quelle colonne répond le mieux à la question posée ?				
	très nettement la gauche	plutôt la gauche	plutôt la droite	très nettement la droite
Moyenne (jeux A et B)	7,27	21,34	**30,0**	**41,39**
Quelle colonne justifie le mieux ses réponses ?				
	très nettement la gauche	plutôt la gauche	plutôt la droite	très nettement la droite
Moyenne (jeux A et B)	8,46	29,74	**42,73**	**19,07**

TABLE 4 – Préférences de présentation des réponses : *gauche* pour le format campagne d'évaluation et *droite* pour Citron.

6 Conclusion

Citron est un système de question-réponse en français capable d'extraire des réponses à des questions à réponses multiples en domaine ouvert à partir de documents provenant du Web. La particularité de Citron est qu'il exploite les structures (énumérations, tableaux) propres à ces documents pour mieux extraire les réponses et éventuellement les agréger pour faire apparaître des critères variants comme la date ou le lieu. Nous nous sommes intéressés ici à la comparaison de performances entre des humains et notre système de question- réponse Citron pour la tâche d'extraction de réponses à des questions à réponses multiples. Nos différentes expériences nous ont permis de confirmer l'hypothèse intuitive qu'un être humain est plus performant que notre système mais à un coût temporel très fort. Elles ont aussi confirmé l'importance pour un système de pouvoir traiter les éléments structuraux comme les tableaux. Mais ces évaluations ont surtout permis de montrer que les utilisateurs produisent eux-aussi des réponses présentant des critères variants et préfèrent la présentation des réponses offerte par Citron, notamment parce qu'elle aide mieux à comprendre la justification des réponses et leur multiplicité.

Références

AÏT-MOKHTAR S., CHANOD J.-P. & ROUX C. (2002). Robustness beyond shallowness : incremental deep parsing. *Natural Language Engineering*, **8**, 121–144.

AYACHE C. (2005). Evaluation en question-réponse, rapport final de la campagne EVALDA. In *Campagne EQUER*.

FALCO M.-H. (2012). Typologie des questions à réponses multiples pour un système de question-réponse. In *RECITAL*, p. 191–204, Grenoble, France.

FALCO M.-H., MORICEAU V. & VILNAT A. (2012). Kitten : a tool for normalizing HTML and extracting its textual content. In *Proceedings of the Eight International Conference on (LREC'12)*, Istanbul, Turkey.

FALCO M.-H., MORICEAU V. & VILNAT A. (2013). Répondre à des questions à réponses multiples : premières expérimentations. In *Conférence francophone en Recherche d'Information et Applications (CORIA)*.

GRAPPY A. (2011). *Validation de réponse dans un système de question-réponse*. PhD thesis, Université Paris-Sud.

HATCHER E., GOSPODNETIC O. & MCCANDLESS M. (2010). *Lucene in action, Second Edition*. Manning.

MORICEAU V. & TANNIER X. (2010). FIDJI : Using Syntax for Validating Answers in Multiple Documents. *Information Retrieval, Special Issue on Focused Information Retrieval*, **13**(5), 507–533.

QUINTARD L., GALIBERT O., ADDA G., GRAU B., LAURENT D., MORICEAU V., ROSSET S., TANNIER X. & VILNAT A. (2010). Question Answering on Web Data : The QA Evaluation in Quaero. In *LREC*, Valletta, Malta.

STRÖTGEN J. & GERTZ M. (2013). Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, **47**(2), 269–298.

TANNIER X. (2012). WebAnnotator, an Annotation Tool for Web Pages. In *LREC 2012*, Istanbul, Turkey.

Centrality Measures for Non-Contextual Graph-Based Unsupervised Single Document Keyword Extraction

Natalie Schluter
Department of Computer Science, School of Technology, Malmö University, Malmö, Sweden
natalie.schluter@mah.se

Abstract. The manner in which keywords fulfill the role of being central to a document is frustratingly still an open question. In this paper, we hope to shed some light on the essence of keywords in scientific articles and thereby motivate the graph-based approach to keyword extraction. We identify the document model captured by the text graph generated as input to a number of centrality metrics, and overview what these metrics say about keywords. In doing so, we achieve state-of-the-art results in unsupervised non-contextual single document keyword extraction.

1 Introduction and Previous Work

Researchers are far from reaching a consensus about what a *keyword extracted* from a document actually is, which may provide part of the explanation for why the keyword extraction task is such a challenge. Some characterisations of keywords are that, with respect to the document at hand, they are important or significant (Mihalcea & Tarau, 2004; Liu *et al.*, 2009a; Wan & Xiao, 2008), salient (Wan *et al.*, 2007), or (less directly) *central*, and that somehow they fulfill the role of little summaries for the document and represent the document. But how keywords fulfill these characteristics, is a question that is still wide open and building a system for a task that is not well-defined is difficult, if not unwise. However, the important role of keywords in tasks such as document indexing for search engines, summarisation, clustering and classification, make this (currently) ill-defined research necessary.

In this paper, we present a study on the essence of keywords in scientific articles and thereby motivate the graph-based approach to keyword extraction. We identify the document model captured by the text graph generated as input to a number of centrality metrics, and overview what these metrics say about keywords. In doing so, we achieve state-of-the-art results in non-contextual single document keyword extraction for the Inspec corpus, an NDCG score of 0.07578 ; we can affirm this, because the systems we compare here in terms of NDCG include re-implementations of the previous state-of-the-art.

We identify two broad types of *single document keyword extraction* (SDKE). *Contextual SDKE* makes use of the document set to which the relevant document belongs, and in which there are similar documents ; other information outside of the document set may also be used in some types of contextual SDKE. *Non-contextual SDKE* makes use of only the relevant document with no other information. The latter does not necessarily make the assumption of independence of documents in general. Non-contextual SDKE is actually important for the case of isolated documents (not part of a document set), as well as for documents for which relevant supplementary information may be non-existent or unreliable. In addition, the study of non-contextual SDKE is a study of baselines in keyword extraction : before constructing complex methods using more information, it is important to understand what we can achieve with the document alone.

This paper presents a study of graph-based unsupervised non-contextual SDKE. Recent studies by Mihalcea & Tarau (2004) (which was the original approach), Rose et al. Rose *et al.* (2010), Litvak, Last & Litvak *et al.* (2013), and Litvak & Last (2008), which also represent the state-of-the-art for the unsupervised version of this task, all use graph-based methods. The superior method (in (Rose *et al.*, 2010), rediscovered in (Litvak *et al.*, 2013)) of the three simply extracts the vertex degree ; the other methods apply more complex algorithms (PageRank (Brin & Page, 1998) is applied in (Mihalcea & Tarau, 2004; Liu *et al.*, 2010; Zhao *et al.*, 2011) and HITS (Kleinberg, 1999) is applied in (Litvak & Last, 2008)). What all three methods have in common, though they do not state that this is their explicit intention, is that they exploit measures of *centrality* of the graph. This paper aims to make these centrality measures explicit and study their appropriateness for the task. Our work will be comparable to that of Mihalcea & Tarau (2004) and Litvak *et al.* (2013), who also worked on unsupervised non-contextual single document keyword extraction for the same dataset, with the latter presenting the previous state-of-the-art. We carry out experiments on the test set from the Inspec abstract corpus (Hulth, 2003) consisting

of 500 abstracts for scientific articles, along with the *uncontrolled* corresponding keywords.

We consider seven well-known measures of centrality, which, following (Borgatti & Everett, 2006) are distributed among the three categories : degree-like centrality, closeness-like centrality, and betweenness-like centrality. In doing so, we hope to either provide motivation, both conceptually (and/or mathematically) and empirically with respect to the appropriateness of these centrality measures. We also give the first non-results oriented motivation for the use of a graph representation of document that provides a basis for which these centrality measures are of interest.

We begin in Section 2 by motivating the document graph model. Then we turn our discussion to centrality measures (Section 3). Finally, we present the experimental results and discussion (Sections 4 and 5).

2 What *is* the graph ?

The graph model in previous work. In non-contextual SDKE, the main motivation for attempting to model the text of a document as a graph is for the use of some ranking algorithm over the textual units of the entire graph, based on some notion of centrality, where the relationships modelled between these units (i.e., the edges or directed edges) are co-occurrence relationships.

(Litvak & Last, 2008) and (Litvak *et al.*, 2013) posit that this reflects linguistic syntax. Certainly, specific linear order textual relations can result from linguistic syntax, to various degrees depending on the language in question, and to a large degree in English specifically. However, it is unclear why syntax alone should be the motivation for the creation of a text graph and its input into a ranking algorithm. What specifically is the role of syntax in determining the most important textual units of a document ?

(Mihalcea & Tarau, 2004), on the other hand, argue that the edges represent connections between concepts (approximated by text units), in terms of their cohesiveness for building up a conceptual context "web" in which a human would understand the text at hand ; in this context web, they explain, some units are more important than others, and they are detectible by their connections with other important concepts, interpreted as "recommendations". They also introduce the notion of a *text surfer*, which we interpret to be the *reader*, over this concept web (PageRank with the concept web as input), as the algorithm for detecting some inherent ranking of units in this web. From these notions, two questions immediately arise. How does a concept (word) recommend another word from mere co-occurrence ? What does it mean for a word to recommend another word ? For the case of the internet, it is clear how these (directed) links form immediate recommendations for given topics. However, this seems to not be as clear in the case of texts.

Sometimes the edges are directed (as in, for example, (Litvak & Last, 2008; Litvak *et al.*, 2013)) and sometimes undirected (as in, for example (Mihalcea & Tarau, 2004)). However, no motivation is given for directed-ness of the relationships based on the nature of the object modelled, over a results related argument from the experiments reported in these studies (they tried both and one type yielded best results), or from some other study (someone else tried both and one type yielded best results).

The graph model in this work : concept-building in communication. In this work, we model text as an *undirected* graph, where vertices are words appearing in the text and edges model text-linear relationships between words (i.e., that they are beside each other in the text) ; vertices representing semantically rich words (approximated here by non-stop-words) are collapsed into a single vertex if they are of the same form and part-of-speech. We follow (Mihalcea & Tarau, 2004), in considering the words of the text as approximations of concepts, but our model motivation differs in two very important respects, as we will now explain.

One essential difference is that we view text from the point of view of *synthesis* (text creation) rather than *analysis* (reading). We posit that (at least) for scientific text, discussing a topic efficiently requires concept coordination. In generating scientific text on a given topic (or given related topics), the "author" may require other concepts to regularly support the discussion (for example, definitions or explanations). We assume that the author is communicating in the most efficient manner possible, and that supporting concepts are named only when absolutely necessary. Moreover, we make the observation that in supporting or defining a concept, textual mention of a topic concept and supporting concepts should occur rather close to each other, in terms of the linear order of concepts (words) in the texts. We therefore approximate these concept support relations by co-occurrence relations, but recognise that these relations are essentially undirected : there is no clear order that should be observed between topic concepts and supporting concepts within a single sentence (or over several sentences for that matter). Note that the network is *not* the meaning of the documentation ; rather it is a *representation of its construction*. Flow through the concept network is seen as *communicative*|concept-building on the part of the author for the reader.

Given this graph, we want to find the most "central" concepts/words and propose these as keywords for the document. However, several notions of centrality can now be applied. A concept can be considered important, because it

1. has first-hand access to many concepts (degree centrality),
2. is very close (and therefore is indirectly used as support) in the network to many distinct concepts (closeness centrality), or
3. is between many concepts and therefore might be often traversed (expressed) in order to build a discussion starting at one concept and ending at another (between-ness centrality).

We examine all these types of centralities using conventional deterministic measures for them, to present how the different centrality measures represent the reality of the gold keyword sets.

Generating the keyword graph. We first carry out sentence detection, tokenisation and part-of-speech tagging on the corpus, using the Stanford POS Tagger (Toutanova *et al.*, 2003). We remove all punctuation from individual sentences. However our sentences are segmented (unlike, for example in (Mihalcea & Tarau, 2004)) ; so we take sentence terminating punctuation into account (like (Litvak & Last, 2008)). We also use a stop-word list to create two different types of graphs, both of which can be constructed in time linear in the length of the document.

1. The **reduced** adjacency graph, which is constructed from the text stripped of stop-words. An edge between two words is added to the graph if these two words are adjacent in the text. Two word vertices decorated with words of the same form and (first letter of) part-of-speech are collapsed into a single vertex. This method follows that of (Liu *et al.*, 2009a). Note that we do not filter out words of a certain part-of-speech tag, unlike (Liu *et al.*, 2009b; Mihalcea & Tarau, 2004; Litvak & Last, 2008) as a preprocessing step.
2. The **full** adjacency graph, which is constructed from the full unstriped text. This time, however, word vertices that are not stop-words of the same form and part-of-speech are collapsed into a single vertex, whereas words that are stop-words are not. So here the text graphs should be seen as having two types of nodes corresponding to (1) candidate keyword parts for the document, and (2) stop words.

The point of using these two separate types of graphs is two-fold : (1) to test if we can eliminate a pre-processing step using by using different measures, and (2) to examine the behaviour of the different centrality measures on the denser (reduced) and sparser (full) graphs.

For Example (1), abstract 1939 from the Test File in the Inspec corpus, first used as an example in (Mihalcea & Tarau, 2004), we give the generated reduced adjacency text graph and full adjacency text graph in Figures 1 and 2, respectively.

(1) Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

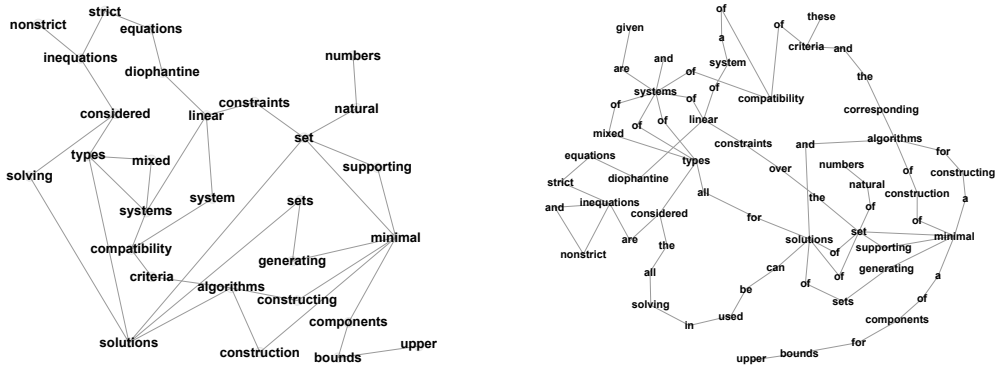


FIGURE 1 – Reduced (left) and full (right) adjacency graph for Example (1).

In this study and to maintain the *unsupervised* characterisation of the study, we use the MySQL stop-word list. [1]

3 Centrality

We hypothesise that keywords are the most "central" word vertices in a document's text graph, under some definition of centrality, which can vaguely be understood to be a summary of a node's involvement in or contribution to the cohesiveness

1. http://dev.mysql.com/doc/refman/5.5/en/fulltext-stopwords.html

of the network (Borgatti & Everett, 2006). We now describe the deterministic centrality measures that we experimented with for the keyword extraction task presented in this paper.

Degree-like centrality measures. The *degree centrality* of a vertex v in a graph G, $C_D(v)$ is simply its degree $deg(v)$. Within the context of text graphs, this is a measure of how much of a first-hand support a text vertex (concept) is for other text vertices (concepts). Previous to the work presented in this paper, systems based on this measure were state-of-the-art (Litvak *et al.*, 2013; Rose *et al.*, 2010).

Eigenvector centrality is generally considered to be a measure of the "influence" of a node in a graph : the more central a node is, the more central its neighbours are and so forth. In other word, vertices are important, because they have first-hand access to many other important vertices. In the context of our concept graph approximated by the text graph, the more support a concept is provided, the more total conceptual support is offered to further concepts supported by it. The output of the PageRank algorithm is meant to be a randomised variant of this measure for *directed* graphs (Page *et al.*, 1999). In fact, this ranking is also theoretically the output of the HITS algorithm (for *directed* graphs) upon convergence (provided all eigenvalues are distinct) (Kleinberg, 1999).

To calculate the eigenvector centrality of a node $v_i \in V(G)$, $C_{EI}(v_i)$, one finds the principal eigenvector of the adjacency matrix for the graph. The ith entry in this vector is $C_{EI}(v_i)$. From this definition, it is clear that the measure cannot be used on disconnected graphs. To surmount this difficulty, we use the PageRank "teleportation trick", transforming the input graph into a complete graph, simply incrementing the weight of all possible edges by 1.

Closeness centrality measures. *Closeness centrality* measures account for the distance of a node to all others. For computational efficiency, they consider the set of all shortest distances of a vertex x to all other nodes : $\{d(x,v) : v \in V(G)\}$. These measures are the least robust of all measures used in the following sense : for a disconnected graph, this sum is infinite for all vertices ; but we use the suggestion of (Dangalchev, 2006) for disconnected graphs, taking the limit in infinite calculations, so the distance between disconnected nodes is infinite and the reciprocal of this is just zero.

The *closeness centrality* $C_C(x)$ of a vertex x is defined as $C_C(x) := \frac{1}{\sum_{v \in V(G)} d(x,v)}$. So, the longer the distance to other word vertices (concepts), the less central the word vertex (concept) can considered. However, with this definition of closeness centrality, one cannot differentiate words that are close and far to equal numbers of nodes from those nodes that are generally close-ish to all other nodes. This is the motivation behind the *eccentricity* measure $C_{ECC}(x)$ of a vertex, which is defined as $C_{ECC}(x) := \frac{1}{\max_{v \in V(G)} d(x,v)}$.

Betweenness centrality measures. The *betweenness centrality* of a vertex quantifies how often a node acts as a bridge along the shortest path between two other nodes. In the context of our text graph, the betweenness centrality can be seen as a measure of how the presentation of a scientific subject must employ a given word (concept) as support when moving the discussion between two different concepts. We consider three different betweenness centrality measures.

The *(normalised) betweenness centrality* $C_B(x)$ for vertex x is defined as $C_B(x) := \sum_{s \in V(G)} \sum_{t \in V(G)} \frac{\sigma_{st}(x)}{\sigma_{st}}$, where σ_{st} is the number of shortest paths between nodes s and t.

$C_B(x)$ gives more weight to pairs of vertices at a larger distance from each other. If one wishes to consider all shortest paths to contribute the same weight, one approach is to normalise by the shortest distance between s and t, which yields *length-scaled betweenness centrality*, $C_{LSB}(x) := \sum_{s \in V(G)} \sum_{t \in V(G)} \frac{\sigma_{st}(x)}{d(s,t)\sigma_{st}}$.

Finally, the *distance-weighted fragmentation* $C_{DWF}(x)$ of vertex x measures the fragmentation of a graph if we took x out of it and is defined as $C_{DWF}(x) := C_{DWF}(G - x) - C_{DWF}(G)$, where $C_{DWF}(G) := 1 - \frac{2\sum_{i \neq j} \frac{1}{d(i,j)}}{n(n-1)}$.

Note that $G - x$ (the graph obtained from G by removing vertex x and any edges incident to x) should be more fragmented than G. (We also shift all scores, so that they are positive.)

Ranked keywords. In Table 1, we give the top ranked seven words output by the degree centrality, eigenvector centrality, closeness centrality and betweenness centrality for Example (1).

$C_D(v)$	systems,	set,	minimal,	solutions,	types,	linear,	inequations
$C_{EI}(v)$	systems,	set,	minimal,	solutions,	types,	linear,	algorithms
$C_C(v)$	set,	solutions,	constraints,	minimal,	algorithms,	generating,	sets
$C_B(v)$	compatibility,	systems,	linear,	constraints,	set,	natural,	numbers

TABLE 1 – Top seven words from ranked lists for Example (1) across a selection of centrality measures, for the full text adjacency graph.

4 Evaluation

We carry out similar post-processing to (Mihalcea & Tarau, 2004). That is, sequences of adjacent keywords from the text are possibly collapsed into a multi-word keyword, depending on their scores. We score a multi-unit keyword by the average (**ave**) score of words they are composed with. This yields a candidate list where there may be unit overlaps in keywords. We therefore test an extra post-processing step which keeps only the keyword with the highest score among two overlapping keywords (this corresponds to **ave-excl** in Table 4). Ties are broken with a preference for longer keywords ; moreover, the proposed keywords must not start or end with a stop-word, and must be grounded in a noun (i.e., the rightmost word of a multi-word keyword must be a noun). Keywords consisting of at most three words are considered.

Results are evaluated using average standard Normalised Discounted Cumulative Gain (NDCG) which is mathematically proven to distinguish between ranking systems that are sufficiently different from each other (Wang *et al.*, 2013). Recall that NDCG is carried out on the *entire* ranked list and not on some top-n items ; System A is considered superior to System B if the positives (correct keywords) are generally ranked higher by System A than by System B according to the NDCG metric.

We see that there is the best performing system uses C_{LSB} (with **(ave-excl)-full**). Moreover, we observe that the post-processing step which does not allow overlapping keywords in the candidate list performs better across all measures ; this is probably because "close duplicates" that would otherwise dilute the higher ranking positions are excluded.

We also consider that the use of the set-based evaluation metrics of precision, recall, and f-score are misleading for rank-based systems, but may be informative as a means of error analysis when considering best parameter performance : poor system performance can possibly be explained by a system reaching its optimal f-score too early (for example, $n = 1$), or too late (for example, $n = 50$) in the ranked list. The best parameter f-scored system is C_B**-(ave)-full** occurs at $n = 9$, with only half of the list being true positives.

The eigenvector centrality measure is equivalent to PageRank and we test the degree centrality measure. These are the two methods that have previously been tested for non-contextual SDKE, in (Mihalcea & Tarau, 2004) and (Rose *et al.*, 2010) respectively. The length-scaled betweenness centrality measure outscores both of these measures in NDCG.

In terms of graph structure, we see that in general the full graph is preferred, which attests to the robustness of the measures and their preference for as much information as possible in the graph.

measure	post-proc	graph	n	precision	recall	f-measure	NDCG
Degree Centrality	ave	reduced	3	0.667	0.25	0.363	0.06064
		full	17	0.353	0.75	0.48	0.06115
	ave-excl	reduced	13	0.308	0.5	0.381	0.05982
		full	14	0.357	0.625	0.455	0.05830
Eigenvector Centrality	ave	reduced	4	0.5	0.25	0.333	0.06192
		full	5	0.4	0.25	0.308	0.06209
	ave-excl	reduced	4	0.5	0.25	0.333	0.05976
		full	5	0.4	0.25	0.308	0.05778
Eccentricity	ave	reduced	6	0.333	0.25	0.286	0.04264
		full	16	0.25	0.5	0.333	0.04535
	ave-excl	reduced	10	0.2	0.25	0.222	0.04769
		full	2	0.5	0.125	0.2	0.04692
Closeness Centrality	ave	reduced	6	0.333	0.25	0.286	0.04264
		full	16	0.25	0.5	0.333	0.04535
	ave-excl	reduced	10	0.2	0.25	0.222	0.04769
		full	2	0.5	0.125	0.2	0.04692
Distance Weighted Fragmentation	ave	reduced	3	0.667	0.25	0.364	0.06204
		full	6	0.5	0.375	0.429	0.06094
	ave-excl	reduced	3	0.667	0.25	0.364	0.05995
		full	6	0.5	0.375	0.429	0.05743
Betweenness Centrality	ave	reduced	3	1.0	0.375	0.545	0.06352
		full	9	0.556	0.625	**0.588**	0.06358
	ave-excl	reduced	3	1.0	0.375	0.545	0.06331
		full	3	1.0	0.375	0.545	0.06334
Length-Scaled Betweenness Centrality	ave	reduced	6	0.667	0.5	0.571	0.06660
		full	3	1	0.375	0.545	0.06658
	ave-excl	reduced	4	0.75	0.375	0.5	**0.07578**
		full	3	1	0.375	0.545	0.07550

TABLE 2 – NDCG and best parameter (n) precision, recall and f-score for all centrality measures tested.

5 Future Work

We have presented a study on the essence of keywords in scientific text, showing that firm understanding of the "flavour" of centrality we are trying to predict is essential in keyword extraction tasks. Some open questions remain. For instance, how robust are these measures on large document graphs ? Also, with larger graphs, time and space becomes an issue : how can these measures be efficiently computed in general ? Large documents pose the next challenge.

Références

BORGATTI S. P. & EVERETT M. G. (2006). A graph-theoretic perspective on centrality. *Social Networks*, **28**(4), 466 – 484.

BRIN S. & PAGE L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*.

DANGALCHEV C. (2006). Residual closeness in networks. *Physica A : Statistical Mechanics and its Applications*, **365**(2), 556–564.

HULTH A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, p. 216–223, Stroudsburg, PA, USA : Association for Computational Linguistics.

KLEINBERG J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, **46**(5), 604–632.

LITVAK M. & LAST M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, p. 17–24, Stroudsburg, PA, USA : Association for Computational Linguistics.

LITVAK M., LAST M. & KANDEL A. (2013). Degext : a language-independent keyphrase extractor. *J. Ambient Intelligence and Humanized Computing*, **4**(3), 377–387.

LIU F., PENNELL D., LIU F. & LIU Y. (2009a). Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, p. 620–628, Stroudsburg, PA, USA : Association for Computational Linguistics.

LIU Z., HUANG W., ZHENG Y. & SUN M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, p. 366–376, Stroudsburg, PA, USA : Association for Computational Linguistics.

LIU Z., LI P., ZHENG Y. & SUN M. (2009b). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1 - Volume 1*, EMNLP '09, p. 257–266, Stroudsburg, PA, USA : Association for Computational Linguistics.

MIHALCEA R. & TARAU P. (2004). Textrank : Bringing order into text. In *Proceedings of EMNLP*, p. 404–411.

PAGE L., BRIN S., MOTWANI R. & WINOGRAD T. (1999). *The PageRank Citation Ranking : Bringing Order to the Web.* Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

ROSE S., ENGEL D., CRAMER N. & COWLEY W. (2010). *Automatic Keyword Extraction from Individual Documents*, In *Text Mining. Applications and Theory*, p. 1–20. John Wiley and Sons, Ltd.

TOUTANOVA K., KLEIN D., MANNING C. D. & SINGER Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, Stroudsburg, PA, USA : Association for Computational Linguistics.

WAN X. & XIAO J. (2008). Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, p. 855–860 : AAAI Press.

WAN X., YANG J. & XIAO J. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *ACL*.

WANG Y., WANG L., LI Y., HE D. & LIU T.-Y. (2013). A theoretical analysis of ndcg type ranking measures. In *Proceedings of COLT*, p. 25–54.

ZHAO W. X., JIANG J., HE J., SONG Y., ACHANANUPARP P., LIM E.-P. & LI X. (2011). Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies - Volume 1*, HLT '11, p. 379–388, Stroudsburg, PA, USA : Association for Computational Linguistics.

Détection de périodes musicales d'une collection de musique par apprentissage

Rémy Kessler[1] Nicolas Béchet[2] Audrey Laplante[1] Dominic Forest[1]
(1) Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal H3C 3J7, Canada
IRISA, UMR 6074, INSA Rennes
{remy.kessler, audrey.laplante, dominic.forest}@umontreal.ca
nicolas.bechet@irisa.fr

Résumé. Dans ces travaux, nous présentons une approche afin d'étiqueter une large collection de chansons francophones. Nous avons développé une interface utilisant les paroles comme point d'entrée afin d'explorer cette collection de musique avec des filtres en fonction de chaque période musicale. Dans un premier temps, nous avons collecté paroles et métadonnées de différentes sources sur le Web. Nous présentons dans cet article une méthode originale permettant d'attribuer de manière automatique la décennie de sortie des chansons de notre collection. Basée sur un système évalué au cours d'une des campagnes DEFT, l'approche combine fouille de textes et apprentissage supervisé et aborde la problématique comme une tâche de classification multi classes. Nous avons par la suite enrichi le modèle d'un certain nombre de traits supplémentaires tels que les tags sociaux afin d'observer leur influence sur les résultats.

Abstract. In this paper, we present an approach to label a large collection of songs in French by decade. We have developed an information visualization interface that allows users to browse the collection searching for lyrics with musical periods dependent filters. We first harvested lyrics and metadata from various sources on the web. We present in this article an original method to automatically assign the decade of songs for our collection. The original system was developed for a DEFT challenge and combined text mining and machine learning with a multi class approach. We subsequently enriched the model with additional features such as social tags to determine their impact on the results.

Mots-clés : Fouille de textes, apprentissage supervisé, paroles de chansons, tags sociaux.

Keywords: text mining, machine learning, lyrics, social tagging.

1 Introduction

Considérant la grande quantité d'informations textuelles sur la musique pouvant être extraite du Web, ce qui inclut métadonnées diverses et paroles de chanson, de nouvelles avenues pour l'étude de la chanson populaire s'ouvrent aux chercheurs. Au moyen de techniques de fouille de textes, il devient désormais possible de développer des systèmes permettant aux chercheurs d'étendre leur étude des paroles de chansons à de larges corpus. Le projet IMAGE[1] propose d'utiliser les paroles comme point d'entrée afin d'explorer une collection de musique. Dans nos précédents travaux (Kessler *et al.*, 2014a) , nous présentions une approche de fouille de textes ainsi qu'une interface permettant la recherche et l'exploration d'une large collection de chansons francophones sur la base des thématiques abordées dans leurs paroles. Les applications de notre démarche sont nombreuses et variées dans le contexte du Web et de la diffusion de contenu multimédia (analyse de la proximité entre les artistes sur la base des thématiques abordées dans leurs chansons, analyse diachronique, etc.) ainsi qu'auprès des sociologues, musicologues qui étudient la musique populaire. Notre corpus incluant des métadonnées riches sur les chansons qui le composent, nous avons développé des fonctionnalités supplémentaires. Nous avons associé à chaque chanson une période musicale (années 80, 90, etc.) afin de visualiser les chansons en fonction de chaque période. Au cours de ce processus, nous avons dû écarter une grande quantité de chansons de notre corpus initial dans la mesure où nous ne disposions pas d'information concernant leur date de sortie. Afin de compléter les informations contenues dans la collection et d'éviter une tâche fastidieuse d'étiquetage manuel, nous avons décidé d'exploiter le reste de la collection ainsi que les métadonnées dans le but d'étiqueter automatiquement la collection. Après une présentation des travaux réalisés par d'autres chercheurs dans ce domaine, nous présenterons succinctement la méthode de collecte du corpus ainsi que des statistiques. Nous aborderons ensuite les traitements appliqués aux paroles et métadonnées avant de développer notre approche pour détecter automatiquement la décennie de chaque chanson.

[1] Indexation de Musique À Grande Échelle

2 Travaux connexes

Étant donné l'abondance d'informations musicales disponibles en accès libre sur le Web, il n'est pas surprenant de constater qu'un grand nombre de chercheurs aient développé des outils pour collecter et exploiter ces informations dans le repérage de la musique (voir Li *et al.* (2012) pour une revue exhaustive de la littérature sur ce sujet). Ainsi, McKay *et al.* (2010) utilisent les paroles de chansons en combinaison avec d'autres données telles que descripteurs acoustiques et symboliques, contenu culturel tiré du Web afin d'améliorer la classification automatique par genre. Même si l'approche sous forme de sacs de mots reste la plus répandue pour classifier les chansons par genre,Mayer *et al.* (2008) utilisent des caractéristiques additionnelles telles que les rimes ou les catégories grammaticales. Des chercheurs ont également démontré la pertinence d'utiliser les paroles pour réaliser d'autres tâches, notamment pour la détection d'émotions (Hu *et al.*, 2009; Van Zaanen & Kanters, 2010), ou encore dans le but d'établir le degré de similarité entre artistes à l'aide d'une analyse sémantique des paroles (Logan *et al.*, 2004). Müller *et al.* (2007) proposent un moteur de recherche sur les paroles de chanson permettant d'accéder directement à la partie de l'enregistrement audio correspondant à sa requête au moyen d'un alignement automatique entre paroles et bande sonore. L'utilisation des tags sociaux[2] dans le cadre des données musicales est devenue pratique courante afin d'améliorer la description des ressources en ligne. En effet, ils offrent un grand volume d'informations textuelles qui va bien au-delà des métadonnées standards. Levy & Sandler (2008) montrent leur l'efficacité dans un système de recherche tout en soulignant les problèmes de polysémie[3] ou encore de synonymie[4]. De la même façon, Laurier *et al.* (2008) les utilisent afin de constituer une référence pour évaluer leur système tandis que (Trant, 2009) décrit une méthodologie permettant la construction automatique d'une taxonomie hiérarchique en fonction de leurs fréquences. Même si les tags sociaux ont des caractéristiques particulières, nous pensons qu'ils sont une source de données d'apprentissage pertinente. À notre connaissance, il n'existe pas de travaux visant à détecter la période de création de chansons ou d'autres pièces musicales. Néanmoins, l'analyse des informations temporelles est souvent une composante essentielle dans la compréhension de textes et est utile dans un grand nombre d'applications de recherche et organisation d'informations comme le souligne (Alonso, 2008; Kanhabua, 2009). La tâche de variation diachronique a ainsi fait l'objet de deux campagnes d'évaluation DEFT[5]. Différentes approches statistiques et symboliques, en combinaison avec des techniques d'apprentissage, ont été développées par les participants afin de détecter la période de textes journalistiques. Généreux (2010) a ainsi pu mettre en évidence des variations de fréquences ou des mots saillants pour chaque décennie tandis que Raymond & Claveau (2011) ont pris en compte les variations de la ponctuation au cours du temps. Au niveau linguistique, une belle étude de réformes orthographiques a été effectuée par Albert *et al.* (2010), permettant aux auteurs de concevoir des filtres pour chaque période spécifique. Des ressources externes ont aussi été utilisées, telles que des entités d'événements nommés (Monceaux & Tartier, 2010) ou encore une base de données avec la date de naissance de personnages célèbres (García-Fernandez *et al.*, 2011). Même si la sémantique de la musique reste difficile à détecter comme le montre les travaux précédents, nous pensons que les tags sociaux et les paroles de chansons, au travers des styles ou genres musicaux, fournissent des descripteurs pertinents.

3 Constitution d'un corpus de chansons

Dans le domaine musical, le jeu de données le plus imposant est le *Million Song Dataset* (MSD) (Bertin-Mahieux *et al.*, 2011) qui, comme son nom l'indique, contient plus d'un million de chansons. Il est accompagné des paroles de plus de deux cent mille chansons représentées sous forme de matrice sac de mots, le *MusiXmatch Dataset*, lequel n'inclut malheureusement qu'une faible portion de chansons en français. Nous avons donc choisi de constituer notre jeu de données. À l'aide d'une liste de pays francophones, nous avons utilisé les API de différents entrepôts de données musicales (*LyricWiki*, *EchoNest*, *LastFM*, *Paroles.net* et *musiXmatch*,) afin de collecter un grand nombre de données sur chaque chanson. Comme il s'est avéré difficile de trouver des informations fiables sur la langue des chansons, nous avons comparé les paroles collectées à des antidictionnaires dans plusieurs langues (anglais, français, etc.) afin de détecter automatiquement la langue de chaque chanson. Le processus de collecte d'informations est détaillé dans (Kessler *et al.*, 2014b) . Cette méthode nous a permis de collecter les paroles de chanson ainsi que de riches métadonnées, incluant les tags sociaux de Last.fm et diverses données bibliographiques (p. ex. : nom du parolier, du compositeur) couvrant une période de 1950 à nos jours. Comme il est courant pour les artistes de sortir plusieurs versions d'un même titre (single, compilations, live), nous avons dû retirer beaucoup de doublons de notre collection. Pour ces expériences, nous avons extrait un sous-ensemble de 5000

[2]Les tags sociaux sont les mots clés libres que les utilisateurs d'un service attribuent aux ressources (p. ex. : chanson, album ou artiste), notamment dans le but de les repérer par la suite.

[3]le tag « love » peut correspondre à une chanson favorite aussi bien qu'une chanson parlant d'amour

[4]la désignation de musique électronique peut être retrouvée au travers de différents termes tels que « musique électro » ,« musique électronique » ou encore « électro » tout court

[5]DÉfi Fouille de Textes `http://deft.limsi.fr/`

chansons de cette collection. Nous n'avons retenu que des chansons en français pour lesquelles nous avions une date de sortie dans les métadonnées. Nous avons ensuite procédé à un échantillonnage stratifié visant une répartition homogène des chansons par période. Le tableau 1 présente quelques statistiques descriptives de ce sous-ensemble.

	avant prétraitements linguistiques	*après prétraitements linguistiques*
Nombre total de mots	1 166 445	506 082
Nombre total de tags sociaux différents	4 417	1 363
Moyenne de mots par chanson	225,68	100,22
Moyenne de tags sociaux par chanson	1,92	1,73

TAB. 1 – statistiques du sous-ensemble de la collection.

4 Méthodologie

4.1 Filtrages et prétraitements linguistiques

Certains différents prétraitements linguistiques sont effectués : conversion des majuscules en minuscules, retrait des chiffres et des nombres (numériques et/ou textuels), des accents et des symboles (par exemple « $ » ,« # » , « * »). Afin d'éviter l'introduction de bruit dans le modèle, nous utilisons un antidictionnaire composé de verbes auxiliaires et d'autres mots fonctionnels couramment utilisés comme marqueurs discursifs ou comme conjonctions de coordination et/ou de subordination. Nous avons par ailleurs enrichi l'antidictionnaire de termes extraits de la langue populaire du Québec et de France que l'on retrouve fréquemment dans les chansons (p. ex. : « té » (tu es), « chu »(je suis), « c'te » (cette)). Nous appliquons par la suite un processus de lemmatisation simple[6] afin de réduire considérablement les dimensions de l'espace tout en augmentant la fréquence des termes canoniques.

4.2 Une approche par Boosting

Nous avons utilisé l'algorithme *AdaBoost* qui combine les votes pondérés d'une multitude d'apprenants faibles. L'algorithme a été implémenté dans l'outil de classification à large-marge BoosTexter (Schapire & Singer, 2000). Il permet de fournir au classifieur des données numériques, mais également des données textuelles sous forme de n grammes de mots. Nous avons utilisé l'outil IcsiBoost (Favre *et al.*, 2007), qui est l'implémentation open-source de cet outil et qui présente l'avantage de pouvoir fournir un score de confiance pour chaque exemple à classifier. Le système de base pour ces expérimentations a été évalué lors de la campagne DEFT 2010 sur la tâche de variation diachronique comme un des systèmes initiaux (Rk_icsiboost) de (Oger *et al.*, 2010) et sera présenté comme baseline (appelé par la suite *SB*) dans la section 5. Les paramètres d'entrée du système sont les entrées lexicales de chaque document, les paroles des chansons ainsi que le titre avec une représentation sous forme de n grammes de mots dont nous avons fait varié la taille entre 1 et 5. Nous avons par la suite enrichi le modèle d'un certain nombre de traits supplémentaires afin d'observer leur influence sur les résultats. Nous présentons ci-dessous l'ensemble de ces traits regroupés en fonction de leur type :

- Traits *EchoNest* : Il s'agit d'un ensemble de valeurs numériques pour chaque chanson transmise par le site *EchoNest*. Différentes caractéristiques pour chaque chanson sont ainsi ajoutées telles que « danceability », « Hotness », « loudness » ou encore tempo.
- Collaborateurs : Chaque chanson étant associée à des collaborateurs (parolier, compositeur, arrangeur, musiciens, etc.), ceux-ci ont été regroupés dans un champ textuel unique. Nous avons cependant fixé la taille des n grammes à 3 afin de conserver l'association entre le type de collaboration et le nom du collaborateur (par exemple,« parolier Martial Tricoche »).
- Chansons et artistes similaires : Nous avons regroupé dans ces traits l'ensemble des chansons et artistes similaires associés à chaque chanson selon les informations issues des sites *LastFM* et *EchoNest*.
- Tags sociaux : Les tags sociaux associés à chaque chanson ont été regroupés en un champ textuel unique avec une taille de n grammes de 1. Les prétraitements classiques sont appliqués (antidictionnaire et lemmatisation) en français ainsi qu'en anglais étant donné que plusieurs utilisateurs du site Last.fm attribuent des tags dans cette langue, même pour les chansons francophones.

Nous présenterons en section 5 les différents résultats obtenus pour chacun des types de traits ainsi que pour leurs combinaisons.

[6]La lemmatisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculins singulier.

4.2.1 Utilisation de motifs séquentiels

Toujours dans l'idée de détecter des descripteurs additionnels sur la base des informations textuelles à notre disposition, nous avons eu recours à l'utilisation des motifs séquentiels. Nous introduisons dans un premier temps ces derniers avant de présenter la méthode mise en œuvre afin d'exploiter les motifs séquentiels avec des données textuelles.

Définitions. Introduite par (Agrawal & Srikant, 1995), l'extraction de motifs séquentiels est un champ de la fouille de données visant à extraire des régularités dans un jeu de données se présentant sous la forme de séquences. Ce jeu de données est appelé *base de données séquentielles* (SDB). Les auteurs ont introduit la notion de *séquence* notée $s = \langle I_1 \dots I_m \rangle$, comme étant une liste ordonnée d'*itemsets* . Un itemset, noté $I = (i_1 \dots i_n)$ est un ensemble de littéraux appelés *items*. Ainsi, la séquence $\langle (a)\ (a\ b\ c)\ (a\ c)\ (d) \rangle$ est ici composée de quatre itemsets, dont le deuxième itemset comporte trois items a, b et c. Une séquence $S_1 = \langle I_1 \dots I_n \rangle$ est *incluse* dans une séquence $S_2 = \langle I'_1 \dots I'_m \rangle$ si il existe des entiers $1 \leq j_1 < ... < j_n \leq m$ tel que $I_1 \subseteq I'_{j_1}$,..., $I_n \subseteq I'_{j_n}$. On dit alors que S_2 *supporte* S_1 et que S_1 est une *sous-séquence* de S_2. Par exemple, $\langle (a)(a\ c) \rangle \subseteq \langle (a)(a\ b\ c)(a\ c)(d) \rangle$. Le *support* d'un motif séquentiel P (notée $sup(P)$) est le nombre de séquences de la SDB supportant P divisé par le nombre de séquences de la SDB. Finalement, un motif séquentiel fréquent est dit ***maximal*** si il n'existe aucun motif pouvant être inclus dans ce dernier.

Utilisation des motifs pour les textes des chansons. Dans notre cas, chaque chanson est considérée comme une SDB. Ainsi, chaque vers représente une séquence. Les itemsets constituants les séquences sont alors construits à partir des mots de chaque vers de la manière suivante. Pour chaque mot, un lemme et une catégorie grammaticale sont extraits afin de former un itemset[7]. Considérons par exemple le vers suivant extrait d'une chanson : "*Sauver une âme*". La séquence résultante sera alors :$\langle\ (Sauver\ VERBE)\ (un\ PRONOM)\ (âme\ NOM)\ \rangle$.

Afin d'obtenir de nouveaux descripteurs caractérisant les chansons nous procédons alors comme suit : 1) construire les SDB pour chaque chanson, 2) extraire les motifs séquentiels maximaux à partir des SDB, 3) conserver uniquement les k motifs les plus fréquents afin de représenter chaque chanson. Finalement, une mesure de similarité est en cours de réalisation afin de mesurer la proximité entre deux motifs séquentiels ce qui nous permettra de calculer des médoïdes représentant les différentes classes. Toute nouvelle chanson, une fois les motifs séquentiels la caractérisant extraits, pourra alors être comparée à ces médoïdes afin de lui attribuer automatiquement une classe.

5 Expériences

5.1 Protocole expérimental

Afin d'entraîner les classifieurs, une étiquette a été associée à chaque chanson en fonction de sa période musicale (années 80, 90, etc.). Compte tenu du peu de documents pour ces périodes dans le corpus, les chansons des années 50, 60 et 70 ont été ramenées à une étiquette unique, années 50-70. Nous avons évalué cette approche avec une validation croisée classique sur 5 partitions du corpus. Chaque chanson est ainsi répartie aléatoirement dans une des partitions, avec un nombre équilibré de chansons pour chaque période. Nous utilisons 4 sous-corpus en tant que données d'apprentissage (trois sont utilisés en apprentissage tandis que le dernier permet de calibrer le classifieur) et le cinquième corpus est réservé au test : 5 jeux de tests tournants sont ainsi créés. Chacune des expériences a été évaluée sur les corpus de test en utilisant les mesures classiques de Précision, Rappel et F-score des documents bien classés, moyennées sur toutes les classes (avec $\beta = 1$ afin de ne privilégier ni la précision ni le rappel (Goutte & Gaussier, 2005)). Le score final d'un test est ensuite calculé en faisant la macro moyenne de tous les scores obtenus sur chacun des ensembles de test. À titre de comparaison, nous avons inclus la mesure d'évaluation développée par (Grouin *et al.*, 2011) permettant d'évaluer le résultat obtenu sur la base d'une fenêtre de 15 ans autour de l'année de référence.

5.2 Résultats

En observant les résultats du tableau 2, on constate que le système classique (SB) obtient un score de 0,38 en Précision et un F-score de 0,54, ce qui est globalement faible, mais comparable aux résultats de la campagne DEFT, même si le nombre de classes était plus important. L'utilisation d'un antidictionnaire et d'une étape de lemmatisation apporte peu individuellement, mais permet d'améliorer les résultats des autres expériences. Concernant les traits EchoNest, ceux-ci apportent peu de gain, mais nous expliquons cela par le nombre conséquent de documents pour lesquels nous n'avions pas ces données. L'apport des autres traits (COL, ACS) reste aussi faible et peu significatif même si l'on observe que les performances sont conservées lors des combinaisons.
L'étude détaillée des résultats montre qu'en dehors des paroles de chansons, les tags sociaux sont le premier trait pris en compte lors de la prise de décision suivi par les motifs ainsi que le nom des collaborateurs. Néanmoins, une observation

[7] Les lemmes et catégories grammaticales sont extraits en utilisant l'outil Treetagger

	Précision	Rappel	F-score	Mesure DEFT 2011
Sans prétraitements linguistiques				
Système de base (SB)	0,383	0,974	0,550	0,279
SB+traits EchoNest (TE)	0,388	0,954	0,551	0,280
SB+Collaborateurs (COL)	0,387	0,961	0,551	0,277
SB+Tags sociaux (TS)	**0,518**	**0,759**	**0,613**	**0,286**
SB+Artistes/Chansons similaires (ACS)	0,383	0,974	0,550	0,279
SB+TE+TS	0,533	0,752	0,620	0,280
SB+TE+TS+COL	0,511	0,773	0,615	0,280
SB+TE+TS+COL+ACS	**0,511**	**0,773**	**0,615**	**0,281**
Avec prétraitements linguistiques				
Système de base (SB)	0,383	0,969	0,549	0,278
SB+traits EchoNest (TE)	0,388	0,971	0,554	0,282
SB+Collaborateurs (COL)	0,388	0,959	0,553	0,281
SB+Tags sociaux (TS)	**0,541**	**0,758**	**0,629**	**0,284**
SB+Artistes/Chansons similaires (ACS)	0,383	0,969	0,549	0,278
SB+TE+TS	0,533	0,752	0,620	0,280
SB+TE+TS+COL	0,404	0,899	0,556	0,277
SB+TE+TS+COL+ACS	**0,545**	**0,754**	**0,630**	**0,290**

TAB. 2 – Ensemble des résultats obtenus par chaque exécution du système.

plus approfondie des erreurs commises montre que l'utilisation de certains tags sociaux comme « français »ou encore « live » influence de façon négative le système. De même, la fréquence importante du tag social « star »pour de nombreux artistes sur des périodes différentes semble conduire le système à commettre des erreurs lors de l'attribution de la décennie. Même si certains tags à caractère personnel comme « favoritos »apportent peu d'information, d'autres tels que « party 2010 »ou encore « seen in 2013 »permettent au système de déterminer la bonne période. En outre, l'utilisation de tags tels que « Disco francophone » , « 1960s » , « french celtic rap » , ou encore les noms d'artistes permettent de réduire de façon significative les erreurs commises par le système. Différentes expériences complémentaires sont en cours afin d'expliquer et améliorer ces résultats. Les résultats obtenus à l'aide de la mesure d'évaluation de la campagne 2011 confirment ces tendances même si les tâches ne sont pas identiques (détection de décennie versus année exacte) . On notera cependant les scores relativement bas ainsi que la faible variation entre les différentes exécutions du système.

6 Conclusion et perspectives

Dans cet article, nous avons présenté une approche combinant fouille de texte et apprentissage afin de retrouver les périodes musicales manquantes d'une large collection. Plus spécifiquement, l'approche repose d'abord sur l'utilisation d'algorithmes de fouille de textes ainsi que sur des techniques d'apprentissage machine. Notre collection est représentée sous forme de sacs de mots complétés par des traits supplémentaires avant de les classifier à l'aide d'un algorithme de boosting. Les résultats obtenus montrent que l'utilisation des paroles et des tags sociaux sont une source intéressante afin d'identifier les liens entre les différentes périodes musicales fournissant une base d'apprentissage non négligeable. Finalement, les premiers tests basés sur les motifs séquentiels présentent des résultats encourageants, mais nécessitent des expériences complémentaires. Nous prévoyons de continuer à augmenter le corpus afin de confirmer ces hypothèses et envisageons d'autres expériences comme la détection de parolier ou encore de styles musicaux propres à chaque artiste.

Remerciements

Les auteurs tiennent à remercier le Conseil de Recherches en Sciences Humaines (CRSH), qui ont partiellement financé ces travaux, le LIA pour la mise à disposition de ressources, Pierre-François Marteau pour sa collaboration ainsi que Mélanie Couderc pour ses relectures attentives.

Références

AGRAWAL R. & SRIKANT R. (1995). Mining sequential patterns. In *Int. Conf. on Data Engineering* : IEEE.

ALBERT P., BADIN F., DELORME M., DEVOS N., PAPAZOGLOU S. & SIMARD J. (2010). Décennie d'un article de journal par analyse statistique et lexicale. In *Actes DEFT 2010*, p. 85–97, Montréal, Canada.

ALONSO O. R. (2008). *Temporal information retrieval*. PhD thesis, University of California at Davis, Davis, CA, USA. Adviser-Gertz, Michael.

BERTIN-MAHIEUX T., ELLIS D., WHITMAN B. & LAMERE P. (2011). The million Song Dataset. In *ISMIR 2011, Miami, Florida*, p. 591–596.

FAVRE B., HAKKANI-TÜR D. & CUENDET S. (2007). Icsiboost.

GARCÍA-FERNANDEZ A., LIGOZAT A.-L., DINARELLI M. & BERNHARD D. (2011). Méthodes pour l'archéologie linguistique : datation par combinaison d'indices temporels. In *Actes DEFT 2011*.

GÉNÉREUX M. (2010). Classification de textes en comparant les fréquences lexicales. In *Actes DEFT 2010*, p. 57–68, Montréal, Canada.

GOUTTE C. & GAUSSIER E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. *ECIR 2005*, p. 345–359.

GROUIN C., FOREST D., PAROUBEK P. & ZWEIGENBAUM P. (2011). Présentation et résultats du défi fouille de texte DEFT2011 quand un article de presse a t-il été écrit ? *Actes du septième DÉfi Fouille de Textes*, p.3̃.

HU X., DOWNIE J. S. & EHMANN A. F. (2009). Lyric Text Mining in Music Mood Classification. (5,049), 411–416.

KANHABUA N. (2009). Exploiting temporal information in retrieval of archived documents. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, p. 848.

KESSLER R., LAPLANTE A. & FOREST D. (2014a). Encore des mots, toujours des mots : fouille de textes et visualisation de l'information pour l'exploration et l'analyse d'une collection de chansons en français. In *Journées d'Analyse statistique des Données Textuelles (JADT 2014), Paris, (à paraître)*.

KESSLER R., LAPLANTE A. & FOREST D. (2014b). Exploration d'une collection de chansons à partir d'une interface de visualisation basée sur une analyse des paroles. volume EGC, RNTI-E-26, p. 112–123.

LAURIER C., GRIVOLLA J. & HERRERA P. (2008). Multimodal Music Mood Classification Using Audio and Lyrics. In *Machine Learning and Applications, 2008 ICMLA'08*, p. 688–693.

LEVY M. & SANDLER M. (2008). Learning Latent Semantic Models for Music from Social Tags. *Journal of New Music Research*, (2), 137–150.

LI T., MITSUNORI O. & TZANETAKIS G. (2012). *Music Data Mining*, volume 21. CRC Press Llc.

LOGAN B., KOSITSKY A. & MORENO P. (2004). Semantic Analysis of Song Lyrics. In *ICME'04, IEEE International Conference*, p. 827–830.

MAYER R., NEUMAYER R. & RAUBER A. (2008). Rhyme and Style Features for Musical Genre Classification by Song Lyrics. In *ISMIR 2008*, p. 337–342.

MCKAY C., BURGOYNE J. A., HOCKMAN J., SMITH J., VIGLIENSONI G. & FUJINAGA I. (2010). Evaluating the Genre Classification Performance of Lyrical Features relative to Audio, Symbolic and Cultural Features. In *ISMIR 2010*, volume 10, p. 213–218.

MONCEAUX L. & TARTIER A. (2010). Utilisation d'outils linguistiques pour trouver la date ou l'origine d'un fragment textuel. In *Actes DEFT 2010*, p. 21–34, Montréal, Canada.

MÜLLER M., KURTH F., DAMM D., FREMEREY C. & CLAUSEN M. (2007). Lyrics-Based Audio Retrieval and Multimodal Navigation in Music Collections. In *Research and Advanced Technology for Digital Libraries*, p. 112–123.

OGER S., ROUVIER M., CAMELIN N., KESSLER R., LEFÈVRE F. & TORRES-MORENO J. M. (2010). Système du LIA pour la campagne DEFT'10 : datation et localisation d'articles de presse francophones. In *Actes DEFT 2010*, p. 69–83, Montréal, Canada.

RAYMOND C. & CLAVEAU V. (2011). Participation de l'IRISA à DEFT2011 : expériences avec des approches d'apprentissage supervisé et non supervisé. In *Actes DEFT 2011*.

SCHAPIRE R. E. & SINGER Y. (2000). BoosTexter : A boosting-based system for text categorization. *Machine Learning*, p. 135–168.

TRANT J. (2009). Studying social tagging and folksonomy : A review and framework.

VAN ZAANEN M. & KANTERS P. (2010). Automatic Mood Classification using tf* idf based on Lyrics. In *Proceedings of the 11th Int. Conf. on Music Information Retrieval*, p. 75–80.

AMESURE: une plateforme de lisibilité pour les textes administratifs

Thomas François[1] Laetitia Brouwers[1] Hubert Naets[1] Cédrick Fairon[1]
(1) CENTAL, IL&C, UCLouvain
1, Place Blaise Pascal, 1348 Louvain-la-Neuve
{thomas.francois;laetitia.brouwers;hubert.naets;cedrick.fairon}@uclouvain.be

Résumé. Cet article présente une plateforme dédiée à l'évaluation de la difficulté des textes administratifs, dans un but d'aide à la rédaction. La plateforme propose d'une part une formule de lisibilité spécialisée pour les textes administratifs, dont la conception repose sur une nouvelle méthode d'annotation. Le modèle classe correctement 58% des textes sur une échelle à 5 niveaux et ne commet d'erreurs graves que dans 9% des cas. La plateforme propose d'autre part un diagnostic plus précis des difficultés spécifiques d'un texte, sous la forme d'indicateurs numériques, mais aussi d'une localisation de ces difficultés directement dans le document.

Abstract. This paper presents a platform aiming to assess the difficulty of administrative texts, mostly for editorial assistance purposes. The platform first offers a readability formula specialized for administrative texts, the development of which required the design of a dedicated annotation procedure. The resulting model correctly classifies 58% of the texts on a 5-levels scale and commits serious misclassifications in only 9% of the cases. Moreover, the platform offers a more accurate diagnosis of the difficulty of a text in the form of numerical indicators corresponding to various textual characteristics. It also locates specific local difficulties directly in the text.

Mots-clés : formule de lisibilité, textes administratifs, aide à la rédaction.

Keywords: readability formula, administrative texts, editorial assistance.

1 Introduction

Les textes produits par les différentes administrations d'un État sont souvent perçus comme des lectures peu enthousiasmantes et particulièrement sujettes à des problèmes d'interprétation ou de compréhension, en particulier auprès de ceux que l'on peut qualifier de lecteurs faibles. D'après les dernières enquêtes PISA (De Coster *et al.*, 2011), cette catégorie engloberait presque 20% de lecteurs en moyenne en Europe. Ces problèmes de compréhension sont d'autant plus critiques dans le cas de textes administratifs que ceux-ci sont relatifs à des questions pratiques, voire parfois vitales pour un individu.

Il n'est donc pas surprenant que les administrations de divers pays aient mis en place des initiatives visant à rendre plus accessible les documents qu'elles sont amenées à produire. Plusieurs guides de rédaction simple ont ainsi vu le jour au Québec (Gouvernement du Québec, 2006), en France, en Belgique (Ministère de la Communauté française de Belgique, 2010) et au sein de l'Union européenne (Union européenne, 2011). Ils ont pour but d'orienter les personnes qui rédigent un texte administratif. Ils soulignent notamment l'importance d'écrire clairement en évitant l'ambiguïté, l'excès de détails qui finissent par occulter l'information principale, une absence de structure textuelle ou une structuration non chronologique. Ils recommandent également de mettre le lecteur au premier plan en s'adressant directement à lui, en utilisant un vocabulaire non spécialisé et en organisant l'information selon ses besoins.

Cependant, au vu des efforts répétés des administrations en ce qui concerne l'accessibilité de leurs productions, on peut douter que ces guides de rédaction simple atteignent complètement leurs objectifs. Ils nécessitent non seulement un effort important d'assimilation de la part des rédacteurs, mais également une vigilance constante, sachant qu'il est généralement assez difficile, pour un rédacteur expérimenté, de se rendre compte des difficultés qu'un lecteur non initié peut éprouver face à ses écrits. C'est pourquoi, en collaboration avec le service de la langue de la Fédération Wallonie-Bruxelles (FWB), a été monté un projet de plateforme web qui cherche à évaluer la difficulté de textes administratifs et à proposer des suggestions de simplification.

Cette plateforme vise un double objectif. D'une part, elle propose une estimation globale de la difficulté d'un texte, ce qui demande de disposer d'une formule de lisibilité spécialisée pour les textes administratifs. D'autre part, elle identifie les éléments lexicaux et syntaxiques difficiles du texte, ce qui requiert une analyse plus fine du texte, semblable à ce qui est fait dans le cadre de la simplification automatique des textes. Dans cet article, nous commençons par situer notre projet par rapport à ces deux domaines (lisibilité et simplification automatique) en mettant en évidence les défis liés au contexte particulier envisagé (les textes administratifs) (*cf.* section 2). Ensuite, nous présentons à la section 3 la méthodologie de conception de notre formule de lisibilité spécialisée pour les textes administratifs ainsi qu'une évaluation de celle-ci. Nous détaillons à la section 4 les éléments linguistiques identifiés comme problématiques par le système et la façon dont ceux-ci sont intégrés dans la plateforme. Nous présentons finalement à la section 5 des perspectives de développement pour cette plateforme.

2 Contexte

Depuis longtemps, la problématique de l'évaluation automatique de la difficulté des textes intéresse les chercheurs. Les premiers efforts dans ce domaine remontent au début du 20e siècle. Ils visent à associer automatiquement des lecteurs avec des textes correspondant à leur niveau de lecture à l'aide de formules mathématiques. Ces modèles opèrent sur la base d'un nombre restreint d'informations textuelles (par ex. la longueur des mots, la longueur des phrases ou la proportion de propositions) qui sont combinées à l'aide d'algorithmes statistiques tels que la régression linéaire (par ex. chez Flesch (1948)). Plus récemment, le domaine a connu une informatisation croissante qui a conduit à la création de modèles basés sur des techniques issues de l'intelligence artificielle et qui reposent sur des variables linguistiques plus fines extraites à l'aide de techniques de traitement automatique du langage (TAL). Parmi les nombreux travaux récents, citons Collins-Thompson & Callan (2005), Heilman *et al.* (2008) ou Vajjala & Meurers (2012).

En ce qui concerne le français, les travaux sont nettement moins nombreux. La première formule de lisibilité est due à Kandel & Moles (1958) et constitue une simple adaptation de la formule de Flesch. Par la suite, Henry (1975) propose la première formule spécifique au français langue première (L1), Daoust *et al.* (1996) explorent les applications du TAL en lisibilité du français L1, tandis que François & Fairon (2013) développent une formule également basée sur le TAL pour le français langue étrangère (FLE). Cependant, aucun de ces modèles n'a été conçu spécifiquement pour les textes administratifs. En effet, créer une formule spécialisée pour un public et un type de textes particuliers requiert de disposer d'un corpus de textes dont la difficulté a été évaluée sur un échantillon de la population ciblée. En général, ce type de ressources s'obtient en collectant des textes dans des manuels scolaires organisés en fonction des niveaux d'éducation visés par le modèle. Malheureusement, pour de nombreux contextes et notamment les textes administratifs, cette solution n'est pas envisageable, ce qui freine considérablement la conception de modèles spécialisés.

Au-delà de cette difficulté méthodologique, les formules de lisibilité sont surtout adaptées à des contextes d'utilisation où il s'agit d'évaluer rapidement et globalement un nombre important de textes. Par contre, elles n'offrent pas de diagnostic précis sur les difficultés d'un texte, ce qui relève plutôt du domaine de l'aide à la rédaction ou « writability ». À l'origine, cette approche consistait à appliquer une formule de lisibilité à un texte et à réécrire celui-ci lorsque le résultat obtenu dépassait une valeur donnée. Cette approche a cependant été critiquée, car elle poussait les rédacteurs à écrire en fonction des formules, qui étaient surtout basées sur des critères de surface. Simplifier revenait donc à réduire la longueur des phrases ou des mots, ce qui ne produit pas toujours des textes plus compréhensibles (Davison & Kantor, 1982). Aujourd'hui, le recours à des technologies de TAL permet de proposer une analyse plus fine des difficultés d'un texte et divers systèmes de simplification automatisés sont apparus pour l'anglais (Chandrasekar *et al.*, 1996; Medero, 2011; Siddharthan, 2006) ou le français (Brouwers *et al.*, 2012). Les critiques qui fustigeaient l'aide à la rédaction basée sur la lisibilité semblent désormais moins pertinentes, en particulier parce que la majorité des variables utilisées dans les formules modernes sont fondées sur des connaissances psycholinguistiques. C'est dans ce contexte que nous proposons une première plateforme d'aide à la rédaction pour les textes administratifs pour le français, appelée AMESURE.

3 Une formule de lisibilité pour les textes administratifs

Comme nous l'avons expliqué, la principale difficulté de conception d'un modèle de lisibilité spécialisé pour les textes administratifs est disposer d'un corpus de textes annotés en fonction de leur difficulté de compréhension pour une population donnée. Ce type de corpus n'existant pas à notre connaissance, nous avons pris le parti d'en rassembler un à partir de documents authentiques provenant de la FWB et à destination du grand public. Il s'agit plus précisément de textes

factitifs, c'est-à-dire des documents utilitaires que le citoyen doit lire en vue de réaliser une procédure (ex. : obtenir une prime) ou pour prendre connaissance d'une problématique (ex. : comment déclarer un héritage). Nous avons ainsi collecté 88 textes relevant de huit thématiques couvertes dans les documents de la FWB (sport, culture, enfance, etc.) ainsi que 27 lettres issues de l'administration et destinées à des particuliers.

Ensuite, il a fallu annoter ces textes en fonction de leur difficulté. Pour ce faire, nous avons employé une méthode en deux étapes. Dans un premier temps, les textes ont été annotés à l'aide de la formule de Kandel & Moles (1958), ce qui a permis de les trier en fonction du score de lisibilité qui leur a été attribué (sur une échelle allant de 100 – très facile – à 0 – très difficile) et de les classer en 5 niveaux sur la base des intervalles définis sur cette échelle par de Landsheere (1963). Nous avons alors sélectionné manuellement deux textes au sein de chaque intervalle afin de disposer de documents de complexité aussi diverse que possible. Ces textes ont ensuite été lus par 10 administrés d'âges variés (de 22 à 64 ans) et titulaires, pour la plupart, d'un diplôme universitaire, via une interface d'autoprésentation segmentée [1]. Celle-ci permet de mesurer le temps passé par chaque lecteur sur chaque phrase à l'aide d'un script javascript (qui évite les latences entre le client et le serveur). À la fin de la lecture, deux questions de compréhension sont posées sur le texte, afin de s'assurer qu'une lecture visant à une bonne compréhension du texte a bien été effectuée par le sujet.

Au terme de cette expérience, nous avons obtenu environ dix mesures de vitesse de lecture (ms/mot) pour chacune des phrases issues des dix textes sélectionnés. Cela nous donne 1017 observations, desquelles ont été exclues les données associées à un score inférieur à 50% au test de compréhension (62 données). Après nettoyage, nous avons calculé la vitesse moyenne de lecture des 10 sujets pour chaque phrase, en éliminant la variation spécifique aux sujets à l'aide d'un modèle à effets mixtes (Baayen *et al.*, 2008). Ensuite, une vitesse de lecture moyenne par mot a été calculée au niveau du texte. C'est cette valeur qui constitue l'indicateur de la difficulté du texte (voir Table 1). On observe une bonne corrélation ($r = 0,74$) entre le score obtenu par la formule de Kandel et Moles (score KM) et le temps moyen de lecture par mots (ms / mot) sur nos dix textes. Sur la base des temps de lecture, nous avons défini 5 niveaux de difficulté, qui constitueront l'échelle de référence pour notre formule.

Numéro du texte	Titre du texte	score KM	ms / mot	Niveau
1	La santé de votre enfant	71,3	292,8	1
2	Du couple à la famille (se séparer...)	86,5	304,9	1
3	Des chaussures... Quand les mettre aux pieds ?	81,1	315	2
4	A l'école d'une alimentation saine	75,8	324,4	2
5	L'enseignement spécialisé	46,2	339,7	3
6	Lettre pour la semaine européenne de la vaccination	40,6	340,5	3
7	Cumuls de pensions	57,5	372,3	4
8	Liquidation des subventions ordinaires 2004	15	376,6	4
9	Déclaration de succession	57	379	5
10	Tax shelter	36,5	390	5

TABLE 1 – Classement des textes selon leur temps de lecture moyen.

Pour la seconde étape de l'annotation, nous avons fait appel à des experts issus de l'administration de la FWB. Il a été demandé à chacun d'eux de classer 15 textes parmi les 105 restant à annoter, en fonction des 5 niveaux de difficulté définis lors de la première étape. Pour les assister dans cette tâche, un guide d'annotation leur a été fourni. Il inclut, comme exemples représentatifs de chaque niveau, les textes de la Table 1 numérotés 1, 3, 6, 7 et 10. Au terme de cette annotation, nous avons obtenu 267 avis sur 105 textes, soit une moyenne de 2,5 avis par textes. L'accord inter-annotateurs n'est pas très élevé : l'alpha de Krippendorff (1980) moyen sur les 7 batchs de 15 textes ne dépasse pas 0,37. Notons toutefois que ce type de tâche d'annotation est particulièrement difficile, comme le montre le kappa de $0,398$ obtenu pour une tâche d'annotation de la difficulté de mots dans SemEval 2012 (Specia *et al.*, 2012). Au terme de ce processus d'annotation, chaque texte s'est vu attribuer le niveau moyen des annotations des experts, réduisant les variations personnelles.

Une fois le corpus d'entraînement annoté, la seconde étape de conception de notre formule de lisibilité a consisté à identifier et paramétriser un ensemble de variables linguistiques qui sont liées avec la difficulté des textes, par un lien causal ou simplement corrélationnel. Nous avons ainsi pris en considération 344 variables qui se répartissent en trois grandes classes : lexicale (fréquence lexicale, diversité lexicale, types de voisins orthographiques et longueur des mots), syntaxique (longueur des phrases, formes verbales, ratios de catégories de discours) et sémantique (niveau de personnalisation du texte, densité des idées et cohésion du texte). Ces variables ont été décrites en détail dans François & Fairon (2013).

1. Celle-ci est disponible à l'adresse suivante : `http://amesure-tests.cental.be/`

Enfin, pour entraîner le modèle, nous avons opéré une sélection de variables à l'aide d'un double critère. Tout d'abord, nos variables ont été ordonnées en fonction de leur capacité prédictive, comme le recommandent Guyon & Elisseeff (2003). Nous avons calculé une corrélation de Spearman entre chacune des variables et le niveau de difficulté des textes. Dans un second temps, la meilleure variable au sein de chacune des sous-familles a été retenue. Le but de cette procédure était à la fois de maximiser la quantité d'information à disposition du modèle tout en limitant le nombre de variables reprises dans le modèle, étant donné qu'il doit s'intégrer dans une architecture en temps réel. Au terme de cette étape de sélection, 10 variables ont été retenues. Elles sont présentées à la Table 2 avec la corrélation obtenue pour chacune d'elles. On remarque en particulier que la complexité syntaxique (NMP et CON_PRO) importe davantage que les indices lexicaux dans les textes administratifs, à l'inverse de ce qui a été observé pour les textes de FLE (François & Fairon, 2013).

Nom	**Description de la variable**	**corr.**
ML3	Modèle unigramme lissé basé sur les fréquences de Lexique3 (New *et al.*, 2001)	-0,32
MedianFFFDV	Médiane des fréquences des verbes du texte	-0,47
PAGoug_8000	Proportion de mots absents des 8000 premiers mots de la liste longue de Gougenheim	0,44
TTR_W	Type-Token ratio calculé sur les lemmes	-0,21
PM8	Proportion de mots de plus de 8 lettres	0,40
mean_freqCumNeigh	Moyenne de la distribution des fréquences cumulées des voisins orthographiques	0,50
NMP	Nombre moyen de mots par phrase	0,64
PPasse_C	Proportion de verbes conjugués au participe passé	0,46
CON_PRO	Rapport du nombre de conjonctions sur celui des pronoms	0,54
PP1P2	Nombre de pron. pers. S1 et S2 / mots	-0,42

TABLE 2 – Liste des variables sélectionnées dans le modèle avec leur corrélation de Spearman.

Enfin, un modèle capable d'estimer la difficulté des textes administratifs a été entraîné à l'aide d'une machine à vecteurs de support (SVM) dans laquelle les coefficients ont été régularisés via une norme L2. Une première série d'explorations a permis de sélectionner un *kernel* linéaire, pour lequel le coût (C) a été optimisé par *grid-search*. L'efficacité de ces modèles a été estimée à l'aide d'une procédure de validation croisée à 10 échantillons. Le modèle retenu ($C = 0.05$) atteint une exactitude de 58% pour la classification et une exactitude contiguë [2] de 91%. Il s'agit donc d'un gain de 38% en exactitude et de 39% en exactitude contiguë par rapport au hasard. Pour comparaison, François & Fairon (2013) obtiennent respectivement 50% en exactitude et 80% en exactitude contiguë pour leur modèle sur des textes de FLE. Cela semble indiquer que le modèle se comporte plutôt bien malgré le nombre réduit de données d'entraînement. On notera toutefois qu'il a tendance à rabattre les textes de niveau 1 au niveau 2 et ceux de niveau 5 en 4, à cause du nombre plus réduit de données pour ces catégories.

4 Plateforme d'aide à la rédaction

Cette formule de lisibilité pour les textes administratifs a été intégrée au sein d'une plateforme plus globale d'aide à la rédaction [3]. En plus de cette estimation globale de la difficulté du texte soumis, la plateforme propose actuellement deux types de diagnostics. Le premier fournit une analyse plus détaillée de la complexité d'un texte sur le modèle de Coh-Metrix (Graesser *et al.*, 2004). Il offre ainsi à l'utilisateur une série d'indicateurs mesurant la complexité des différentes dimensions d'un document administratif. Pour l'instant cinq indicateurs ont été retenus : PAGoug_8000, NMP, CON_PRO et PP1P2 (voir Table 2), ainsi qu'une mesure de la cohérence moyenne du texte. Cette dernière est estimée comme le cosinus moyen de toutes les paires de phrases adjacentes du texte représentées sous la forme d'un vecteur de mots dans un espace de dimensions réduites (à l'aide d'une analyse sémantique latente).

Le second diagnostic va plus loin en cherchant à identifier précisément des points d'achoppement probables dans le texte, qu'il s'agisse d'un mot rare ou d'une structure syntaxique considérée comme plus difficile à analyser. C'est à ce niveau que la plateforme rejoint les travaux en simplification de textes, puisque notre approche adopte une approche semblable à ceci près qu'elle s'arrête au niveau de l'identification des éléments à simplifier sans effectuer automatiquement de simplifications. À ce stade de développement de la plateforme, ce module se limite à repérer les mots rares sur la base de leur fréquence, ainsi qu'à identifier trois types de constructions qui peuvent poser des difficultés de lecture. La Figure 1 propose un exemple d'analyse sur un texte court et présente l'interface de la plateforme. Les mots rares sont grasseyés en rouge et les structures complexes soulignées dans une couleur qui dépend de leur type.

2. Il s'agit de la proportion de prédictions qui s'éloignent au plus d'un niveau par rapport au niveau de référence. Son emploi, commun en lisibilité, se justifie par la difficulté qu'ont les experts humains eux-mêmes à accorder leurs jugements.

3. La plateforme est gratuitement accessible à l'adresse suivante : `http://cental.uclouvain.be/amesure/`.

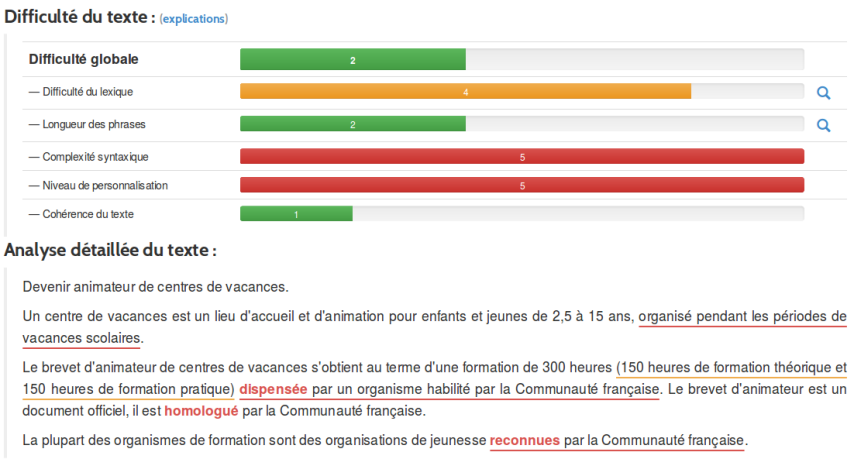


FIGURE 1 – Capture d'écran des résultats de la plateforme AMESURE sur un texte.

La détection des mots rares est réalisée en comparant les logarithmes des fréquences des formes fléchies du texte (tirées de Lexique3 (New *et al.*, 2001)) à un seuil de référence (-14 dans notre exemple), qui pourrait cependant être manipulé par l'utilisateur en fonction du public qu'il vise. Quant aux trois constructions syntaxiques détectées, elles regroupent (1) le passif ; (2) les éléments entre parenthèses et (3) les propositions subordonnées. En effet, lorsque le verbe principal est à la voix passive, la phrase ne suit pas l'ordre SVO auquel s'attend le lecteur, ce qui peut poser davantage de problèmes de compréhension. De même, les éléments entre parenthèses, s'ils sont accumulés, ont tendance à occulter l'information principale qui se retrouve noyée au milieu des informations secondaires. Enfin, les propositions subordonnées contribuent également à allonger et alourdir la phrase, la rendant moins claire. C'est pourquoi de nombreux manuels de rédaction simple (Gouvernement du Québec, 2006; Ministère de la Communauté française de Belgique, 2010; Union européenne, 2011) recommandent de simplifier ce type de structure.

5 Conclusion et perspectives

Cet article a présenté AMESURE, une plateforme dédiée à l'évaluation de la difficulté des textes administratifs, dont le but principal est d'aider les rédacteurs de ce type de textes à produire des documents plus accessibles. La principale contribution de cet article consiste en une formule de lisibilité spécialisée pour les textes administratifs, dont les performances sont comparables à celles d'autres modèles actuels pour le français, malgré un nombre réduit de textes pour l'entraînement. Cela a été rendu possible grâce à une technique d'annotation innovante dans le domaine de la lisibilité qui se base sur la mesure de la vitesse de lecture moyenne d'un groupe de lecteurs. À l'aide de celle-ci, nous avons défini une échelle de difficulté et un guide d'annotation ayant ensuite servi à un groupe d'experts à annoter le reste du corpus. Cette technique pourrait être réutilisée pour entraîner d'autres modèles de lisibilité spécifiques lorsque, comme dans notre cas, il n'existe pas de textes gradués disponibles.

Par ailleurs, cette plateforme intègre divers indicateurs des difficultés présentes dans les textes administratifs relevant de plusieurs dimensions : complexité lexicale du texte, complexité de ses structures syntaxiques, taux de cohérence, etc. Par ailleurs, un diagnostic est également fourni concernant un certain nombre de points d'achoppement relatifs à la rareté du vocabulaire employé et à des structures syntaxiques considérées comme potentiellement problématiques.

À notre connaissance, cette plateforme constitue le premier outil librement accessible pour le français qui offre à la fois une évaluation de la lisibilité d'un texte et un diagnostic plus précis. Il s'agit cependant d'une première étape et les diagnostics, en particulier, pourraient être étoffés. Tout d'abord, on peut imaginer d'autres méthodes pour repérer les mots difficiles, soit en se focalisant sur les termes techniques à l'aide d'outils d'extraction terminologiques, soit en évaluant la difficulté d'un mot à partir de plusieurs critères (et pas seulement de la fréquence) à l'instar de ce qui est proposé par Gala *et al.* (2013). En ce qui concerne les structures syntaxiques complexes, nous comptons également repérer davantage de structures problématiques, telles que les phrases négatives et à double négation, les interrogatives indirectes, les phrases qui ne respectent pas l'ordre sujet-verbe-objet (outre la forme passive), les clivées, les compléments circonstanciels, etc.

Références

BAAYEN R. H., DAVIDSON D. J. & BATES D. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of memory and language*, **59**(4), 390–412.

BROUWERS L., BERNHARD D., LIGOZAT A.-L. & FRANÇOIS T. (2012). Simplification syntaxique de phrases pour le français. In *Actes de la Conférence Conjointe JEP-TALN-RECITAL*, p. 211–224.

CHANDRASEKAR R., DORAN C. & SRINIVAS B. (1996). Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational Linguistics*, volume 2, p. 1041–1044.

COLLINS-THOMPSON K. & CALLAN J. (2005). Predicting reading difficulty with statistical language models. *Journal of the American Society for Information Science and Technology*, **56**(13), 1448–1462.

DAOUST F., LAROCHE L. & OUELLET L. (1996). SATO-CALIBRAGE : Présentation d'un outil d'assistance au choix et à la rédaction de textes pour l'enseignement. *Revue québécoise de linguistique*, **25**(1), 205–234.

DAVISON A. & KANTOR R. (1982). On the failure of readability formulas to define readable texts : A case study from adaptations. *Reading Research Quarterly*, **17**(2), 187–209.

DE COSTER I., BAIDAK N., MOTIEJUNAITE A. & NOORANI S. (2011). *Teaching Reading in Europe : Contexts, Policies and Practices*. Rapport interne, Education, Audiovisual and Culture Executive Agency, European Commission.

DE LANDSHEERE G. (1963). Pour une application des tests de lisibilité de Flesch à la langue française. *Le Travail Humain*, **26**, 141–154.

FLESCH R. (1948). A new readability yardstick. *Journal of Applied Psychology*, **32**(3), 221–233.

FRANÇOIS T. & FAIRON C. (2013). Les apports du TAL à la lisibilité du français langue étrangère. *Traitement Automatique des Langues (TAL),*, **54**(1), 171–202.

GALA N., FRANÇOIS T. & FAIRON C. (2013). Towards a french lexicon with difficulty measures : Nlp helping to bridge the gap between traditional dictionaries and specialized lexicons. In *Electronic lexicography in the 21st century : thinking outside the paper (eLex2013)*, p. 132–151.

GOUVERNEMENT DU QUÉBEC (2006). *Rédiger simplement - Principes et recommandations pour une langue administrative de qualité* . Québec : Bibliothèques et archives nationales du Québec.

GRAESSER A., MCNAMARA D., LOUWERSE M. & CAI Z. (2004). Coh-Metrix : Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, **36**(2), 193–202.

GUYON I. & ELISSEEFF A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, **3**, 1157–1182.

HEILMAN M., COLLINS-THOMPSON K. & ESKENAZI M. (2008). An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, p. 1–8.

HENRY G. (1975). *Comment mesurer la lisibilité*. Bruxelles : Labor.

KANDEL L. & MOLES A. (1958). Application de l'indice de Flesch à la langue française. *Cahiers Études de Radio-Télévision*, **19**, 253–274.

KRIPPENDORFF K. (1980). *Content analysis : An introduction to its methodology*. Beverly Hills, CA : Sage.

MEDERO, J.AND OSTENDORF M. (2011). Identifying targets for syntactic simplification. In *Proceedings of the SLaTE 2011 workshop*.

MINISTÈRE DE LA COMMUNAUTÉ FRANÇAISE DE BELGIQUE (2010). *Écrire pour être lu - Comment rédiger des textes administratifs faciles à comprendre ?* Bruxelles : Ingber, Damar.

NEW B., PALLIER C., FERRAND L. & MATOS R. (2001). Une base de données lexicales du français contemporain sur internet : LEXIQUE. *L'Année Psychologique*, **101**, 447–462.

SIDDHARTHAN A. (2006). Syntactic simplification and text cohesion. *Research on Language and Computation*, **4**(1), 77–109.

SPECIA L., JAUHAR S. K. & MIHALCEA R. (2012). Semeval-2012 task 1 : English lexical simplification. In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, p. 347–355.

UNION EUROPÉENNE (2011). *Rédiger clairement*. Luxembourg : Office des publications de l'Union européenne.

VAJJALA S. & MEURERS D. (2012). On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, p. 163–173.

Décomposition des « hash tags » pour l'amélioration de la classification en polarité des « tweets »

Caroline Brun[1], Claude Roux[1]
(1) XRCE, 6, chemin de Maupertuis, 38240 Meylan
caroline.brun@xerox.xrce.com, claude.roux@xerox.xrce.com,

Résumé. Les « mots dièses» ou « hash tags » sont le moyen naturel de lier entre eux différents tweets. Certains « hash tags » sont en fait de petites phrases dont la décomposition peut se révéler particulièrement utile lors d'une analyse d'opinion des tweets. Nous allons montrer dans cet article comment l'on peut automatiser cette décomposition et cette analyse de façon à améliorer la détection de la polarité des tweets.

Abstract. Hash tags are the natural way through which tweets are linked to each other. Some of these hash tags are actually little sentences, whose decompositions can prove quite useful when mining opinions from tweets. We will show in this article how we can automatically detect the inner polarity of these hash tags, through their decomposition and analysis.

Mots-clés : hash tag, tweet, analyse d'opinion, TAL
Keywords: hash tag, tweet, opinion mining, NLP.

1 Introduction

Twitter est devenu en quelques années le service de mico-blogging le plus populaire sur Internet. Les utilisateurs, appelés aussi « Tweetos » peuvent envoyer sur une plate-forme de partage de petits messages d'au plus 140 caractères, les tweets. De nombreuses expériences ont montré que ces « tweets » étaient souvent un très bon indicateur des choses à venir. Ainsi, [Asur et al. 2010] ont découvert que l'on pouvait prévoir le succès d'un film uniquement sur la base des tweets envoyé à son sujet. Les tweets jouent aussi un rôle dans les plans marketing des entreprises pour détecter comment leurs produits sont perçus par leurs clients. Les hash tags (ou hash tags) en particulier, jouent un rôle spécifique pour assurer un liant entre tous ces messages. Ces « hash tags » sont des métadonnées, des annotations libres définies par les utilisateurs qui servent à marquer l'appartenance d'un message à un domaine particulier, de construire un canal implicite de communication. [Wang et al. 2011] définit en particulier trois familles de hash tags :

- Sujet : hash tags utilisés pour définir grossièrement un sujet particulier : #Sarkozy, #LeDebat… ;
- Sentiment : #Idiot , #Deception, qui indiquent essentiellement une tonalité ;
- Sentiment-Sujet : hash tags, qui recoupent à la fois les sentiments et un sujet particulier : #ViveHollande, #SarkoOnTaime ;

De fait, comme les hash tags sont un élément essentiel des tweets, la plupart des systèmes d'analyse d'opinion cherchent à les incorporer dans leur calcul. [Davidov et al 2010] montre, par exemple, comment l'on peut améliorer les techniques standards de classification supervisée en intégrant la polarité des hash tags les plus fréquents comme paramètre, polarité qui est assignée ici manuellement. [Kouloumpis et al. 2011] ont aussi employé une méthode similaire, mais ont rajouté les émoticons dans la détection de la polarité des tweets.

Cependant, définir à la main la polarité des hash tags n'a rien d'une aventure exaltante et peut se révéler plutôt coûteuse comme opération. [Dave et al. 2012] n'utilise la polarité des hash tags qu'à la condition qu'ils appartiennent à une liste connue à l'avance de mots tels que : #efficace, #nul, #incapable, #visionnaire etc. Mais, malheureusement, ils perdent là la majorité des expressions à mots multiples, dont pourtant raffolent les Tweetos. Pourtant, toutes ces expériences prouvent combien l'utilisation des hash tags peut se révéler précieuse dans l'évaluation de messages dont la taille s'avère souvent trop courte pour que les méthodes traditionnelles fonctionnent de manière optimale. Les hash tags sont par ailleurs souvent la clef pour déterminer l'ironie ou l'humour dans un message donné. Or, ils se révèlent particulièrement difficiles à analyser. Ils peuvent être aussi bien des noms propres, des noms de lieux ou des phrases complètes sans souvent le moindre indice sur leur construction interne.

Dans ce papier, nous proposons une méthode qui permet de détecter automatiquement la polarité d'un hash tag, en appliquant tout d'abord des méthodes de décomposition pour les segmenter, suivie d'une analyse avec une grammaire des sentiments pour en détecter la polarité.

2 Le contexte

L'une des tâches de base dans l'extraction d'opinion ou de sentiment est de classifier la polarité d'un document selon plusieurs axes tels que: neutre, positif ou négatif. Il existe de nombreuses approches pour résoudre ce problème. Certains chercheurs classent les documents selon une échelle numérique, associant aux différents mots une valeur négative ou positive selon leur polarité propre. Ils en déduisent ainsi une polarité globale en fonction du nombre de mots positifs ou négatifs présents dans le document. Mais ces approches ont l'inconvénient de mal refléter les opinions contrastées d'un individu donné. Ainsi dans le cas d'un restaurant, il/elle peut apprécier l'ambiance mais critiquer le service ou la cuisine. L'approche à laquelle nous nous intéressons ici consiste donc à détecter ces différents « aspects » au sein d'un document, aspects pour lesquels nous voulons calculer une polarité adaptée, (voir [Hu & Liu 2004], [Liu 2012]). Ce travail nécessite une approche plus fine, le mot « rouge » par exemple peut être positif pour une viande, mais plutôt négatif s'il est appliqué à l'humeur d'un serveur.

2.1 Imagiweb

Nous avons réalisé l'ensemble de nos expériences avec le corpus de tweets du projet Imagiweb. Ce projet financé par l'ANR[1] consiste à étudier l'image de personnalités politiques ou de compagnies telle qu'elle apparaît sur la Toile, à travers des blogs ou des tweets. En particulier, l'une des tâches consiste à analyser les images de N. Sarkozy et de F. Hollande lors de la dernière élection présidentielle en Mai 2012, pour effectuer une analyse d'opinion sur certains aspects de la perception de ces hommes politiques au sein de la population des Tweetos.

Les données ont été extraites automatiquement de Twitter avec des requêtes appropriées par l'un des partenaires du projet. Le corpus ainsi construit correspond à environ 1500 utilisateurs, sur la base de 10 personnalités politiques différentes. De ce corpus ont été isolé environ 20.000 tweets, distribués de façon uniforme entre N. Sarkozy et F. Hollande, qui ont été ensuite annoté manuellement en polarité selon les aspects suivants : apparence physique, projet politique, sens éthique, communication etc. Soit environ 10 catégories, décomposées en 17 sous-cibles au total.

3 Décomposition

Les données dont nous nous servons au sein du projet ImagiWeb comprennent un grand nombre de « hash tags » correspondant à de petites phrases. Le terme « hash » réfère à une méthode d'indexation bien connue en informatique, le « hachage » qui consiste à calculer un index numérique pour une valeur donnée afin de la ranger dans une table de taille fixe. Par exemple, le calcul du reste d'une division « n : d » rend toujours une valeur comprise entre [0..d-1], ce qui permet de ranger dans une table de dimension « d » n'importe quelle valeur.

Un « hash tag » commence toujours par le caractère « # » ce qui permet de le repérer très rapidement dans le flux d'analyse. Ces « hash tags » posent des problèmes très particuliers à l'analyse linguistique. En effet, ils sont considérés comme des mots inconnus et leur sémantique particulière se perd dans le traitement complet. Or, dans un « tweet » dont la longueur ne peut excéder 144 caractères, ignorer les « hash tags » peut conduire à une dégradation très forte de l'interprétation de celui-ci. En effet, les « hash tags » sont souvent employés soit pour véhiculer des informations sur l'auteur du tweet, soit pour répondre à un autre tweet, soit pour introduire un contre-point au message, sous la forme d'une remarque ironique ou amusante.

Voici quelques exemples de « hash tags » trouvés dans nos corpus: #MariagePourTous, #AvecSarkozy, #voteHollande, #placeaupeuple etc.

Les « hash tags » dans nos corpus tombent peu ou prou dans quatre catégories :

a. Les hash tags qui correspondent à un seul mot de la langue
b. Les hash tags qui correspondent à un nom propre

[1] ANR 2012-CORD-002-01

c. Les hash tags qui correspondent à un groupe de mots
d. Les hash tags impossibles à décoder, le plus souvent des acronymes ou des chiffres.

Les deux premières catégories sont très simples à analyser et ne requièrent pas de traitement très lourd. Il suffit de disposer de lexiques adéquats pour les traiter. La dernière catégorie elle contient une information qui pose déjà des problèmes à des humains pour les comprendre, nous n'avons pas essayé de les interpréter plus avant. Nous nous sommes en revanche concentrés sur la catégorie c) pour tenter de découper en mots le contenu. Peu de travaux s'intéressent a la segmentation des « hash tags », citons [Bakliwal et al. 2012] qui détectent les mots de polarité positive ou négative dans un « hash tag », par exemple « happy » dans « #Iamhappy », mais concluent qu'une segmentation serait plus appropriée, par exemple pour pouvoir détecter les négations. Très récemment, [Maynard & Greenwood 2014] segmentent également les «hash tags » pour améliorer la détection des tweets sarcastiques.

3.1 Découpage

Il existe nombre de méthodes pour découper des chaines en unités lexicales, en particulier dans des langues agglutinatives comme l'allemand, où les mots composés sont non seulement nombreux mais peuvent souvent être construits librement. Un mot comme « Geburtstagfest », qui signifie « fête d'anniversaire » est composé de trois segments : Geburt (être né), Tag(le jour) et Fest (fête). Il existe des règles précises qui gouvernent la composition de ces mots, en particulier ici l'utilisation d'un « s » entre Geburt et Tag. [Koen et al 03] proposent différentes méthodes pour redécouper cette chaine en ses racines premières, en se basant sur leurs fréquences au sein d'un corpus ou d'un lexique. Ils obtiennent ainsi une précision de l'ordre de 83%. Notre problème est très proche de celui du découpage d'un mot composé allemand, mais nous avons préféré utiliser des méthodes plus simples pour résoudre notre problème. Ainsi, pour découper les hash tags composés de plusieurs mots, nous avons utilisé différentes techniques selon la nature du « hash tag » lui-même. L'exemple le plus simple est l'utilisation de majuscule au début des mots pour les différencier entre eux. Dans ce cas, il suffit de repérer celles-ci pour le découper : #MariagePourTous donne « mariage pour tous ».

Cette méthode échoue souvent lorsque la chaîne est en minuscule mais contient un nom propre telle que : #avecHollande. On utilise alors une seconde méthode qui consiste à analyser la chaine en deux phases. Dans un premier temps, on analyse la chaîne depuis le début et on tente de repérer la sous-chaîne la plus longue appartenant à nos lexiques. Nous choisissons la chaîne la plus longue pour éviter certains écueils. Par exemple dans le hash tag « #placeaupeuple », on veut éviter d'arrêter le traitement après avoir détecté « peu », mais au contraire aller jusqu'au bout pour isoler « peuple ». On effectue la même opération en partant aussi de la fin. Analyser depuis la fin ou depuis le début ne donne pas forcément les mêmes résultats. Ce travail à priori simple présente une certaine combinatoire parfois difficile à contrôler. Nous travaillons d'ailleurs sur des méthodes plus sophistiquées, basé sur l'utilisation d'automates pour effectuer la reconnaissance des sous-chaînes.

3.2 Evaluation

On évalue ensuite les trois découpages en comptant le nombre de mots inconnus et on garde celui qui offre le score le plus faible. Nous avons appliqué cette méthode sur notre corpus et nous avons obtenu une précision d'environ 80% pour les 1132 « hash tags » extraits de nos 20.000 tweets. Cette précision a été obtenue en vérifiant chacun des découpages produits à la main. Pour information, sur les 1132 hash tags, 524 présentaient une possibilité de découpage (soit 46%), et parmi eux 160 (soit 30% de ces mots composés ou 14% du total) utilisaient une délimitation avec majuscule.

4 Intégration dans un système de détection d'opinion

4.1 Modèle d'une opinion

D'un point de vue formel, notre système de détection d'opinion adopte la représentation d'une opinion selon le modèle proposé par [Liu 2010], où une opinion est un prédicat d'arité 5 de la forme (o_j, f_{jk}, SO_{ijkl}, h_i, t_i) avec:

- o_j est l'objet cible de l'opinion (le concept principal) f_{jk} est un aspect associé à l'objet o_j
- SO_{ijkl} est la valeur (positive ou négative) de l'opinion exprimée par le locuteur h_i à propos de l'aspect f_{jk}
- h_i désigne le locuteur

– t_i est le moment où l'opinion a été exprimée

Notre système d'extraction est basé sur un analyseur syntaxique qui fournit pour chaque énoncé un ensemble de dépendances syntaxiques. Celles-ci sont alors réinterprétées pour produire des relations sémantiques qui nous servent à instancier nos prédicats.

4.2 Détection de l'opinion

Nous utilisons notre analyseur syntaxique comme composant fondamental pour extraire les dépendances syntaxiques profondes à partir desquelles sont construites nos relations sémantiques, à partir desquelles les prédicats sont instanciés. Elles sont encodées sous la forme suivante : OPINION[POLARITE](POLAR-PREDICAT,OPINION-CIBLE), où OPINION est le nom de la relation sémantique, POLARITE un trait associé avec la dépendance, dont les valeurs sont : "POSITIVE" ou "NEGATIVE". POLAR-PREDICAT est l'expression dans l'énoncé portant la polarité de l'opinion et OPINION-CIBLE, la cible de l'opinion.

Ces relations sémantiques sont en fait produites par des règles particulières au sein de notre analyseur robuste, qui combine à cette fin, informations lexicales spécialisées et dépendances syntaxiques. En particulier notre lexique comprend aussi bien des mots du domaine dont la polarité est connue que les hash tags extraits préalablement. Si les lexiques ont été construits en appliquant des techniques de clustering et de classifications sur de larges corpus ([Brun 2012]), les règles sémantiques sont développées manuellement ([Brun 2011]). Ces règles prennent la forme suivante :

If (SUJET(#1[!polarité:!], #2))==> SENTIMENT[polarité](#2,#1)

Cette règle peut se lire de la façon suivante. Si un verbe donné (ici #1) portant une certaine polarité est sujet d'un nom #2, alors on produit une dépendance sémantique SENTIMENT dont la polarité sera celle de ce verbe. (!polarité :! déclenche une percolation du trait à partir du verbe, lequel est alors capturé par la dépendance).

Exemples :

– Ces enchiladas#2 m'émerveillent#1 sans cesse.
– Cette voiture#2 m'a profondément déçue#1.

Environ une centaine de règles ont été ainsi écrites pour couvrir un large éventail de structures identifiées dans les corpus d'apprentissage. Notre système appartient à la même famille que celui de [Kim et Hovy 2006] ou [Wu et al. 2009], qui utilisent aussi des dépendances syntaxiques pour lier la source et la cible des opinions. De fait, avec ce type d'approche, il devient possible d'intégrer le traitement de phénomènes complexes comme la négation par exemple.

4.3 Adaptations aux tweets

Dans une première passe, nous avons détecté tous les hash tags présents dans le corpus. Puis nous avons appliqué notre mécanisme de découpage sur chacun d'entre eux. Comme nous disposons à l'interne d'une grammaire d'opinion, enrichie d'un vocabulaire spécifique au domaine que nous traitons, nous avons pu l'appliquer sur ces décompositions de façon à attribuer à chacun de ces hash tags une polarité positive ou négative. De cette façon, nous avons pu composer un lexique de hash tags, considérés comme des noms, chacun associé avec un trait particulier pour indiquer sa polarité, et éventuellement sa cible.

A partir des 896 hash tags correctement décomposés, nous avons obtenu les résultats suivants :

– 215 hash tags encodant à la fois la polarité et la cible comme: #VotezHollande,#HollandeHonte, #SarkoOnTaime
– 304 hash tags encodant seulement la polarité : #Abruti, #CasseToi
– 377 hash tags encodant seulement le sujet, parmi lesquels 169 sont des entités nommées.

Une fois cette analyse effectuée, nous les avons enregistrés dans un lexique spécialisé sous la forme suivante:

– "#SarkoPipo": noun[negative=+,target="Sarkozy"].
– "#VoteHollande" : noun[positive=+, target="Hollande"].
– "#CasseToi" : noun[negative=+].

Ce lexique a été alors intégré à notre grammaire d'opinion avant d'appliquer celle-ci à notre corpus de tweets. De cette façon, lors du traitement linguistique, les hash tags sont interprétés comme des noms par les couches basses de la grammaire (segmentation et analyse morphologique) puis enrichis avec les informations de polarité provenant de ces nouveaux lexiques spécialisés. De cette façon, les données de polarité introduites par les hash tags peuvent être intégrées dans le calcul de polarité de chacun des aspects dans les tweets.

5 Expériences et résultats

Pour évaluer l'impact de l'intégration de la polarité des hash tags et des cibles dans notre système de détection d'opinion, nous avons procédé à plusieurs expériences de classification sur notre corpus Imagiweb de tweets annotés. Ce corpus comprend donc 3920 tweets dont les opinions sont annotées en polarité (ici positive ou négative) et en cible, comprenant un total de 392 hash tags décomposés. La mesure de performance choisie est l'exactitude de la classification. Nous avons comparé le comportement de notre système sur la détection de la polarité des tweets une première fois sans les hash tags, et une seconde fois en les intégrant. Nous avons utilisés les sorties de notre analyseur pour entrainer un SVM avec différents paramétrages (SVMLight, [Joachims 1999]), de tels classifieurs s'étant montrés performants pour des tâches de classification de textes. Pour chaque expérience, la classification est effectuée sur le même ensemble de données d'entrainement et de test, extraites de façon aléatoire du corpus initial, tandis que les résultats sont calculés via une procédure de validation croisée en 10 découpages différents (ten-fold). Le jeu de test correspond à 10% du corpus initial, et le jeu d'entrainement au 90% restant. Chaque ensemble comprend la même distribution de tweets positifs et négatifs.

- Expérience1 (référence) est basé sur l'approche en sac de mots.
- Expérience 2 (hash tags) rajoute les hash tags et leur polarité
- Expérience 3 (opinions) rajoute les opinions détectées sans les hash tags
- Expérience 4 (opinions+hashtags) intègre les opinions détectées et les hash tags.

L'exactitude moyenne observée lors d'une validation croisée est estimée selon une erreur moyenne quadratique. La table suivante résume le résultat des quatre expériences.

Expériences	Exactitude %
Expérience 1 : sac de mots	80,1
Expérience 2 : hash tag	82,6
Expérience 3 : opinions	80,2
Expérience 4: opinions+hash tags	82,2

Alors que l'utilisation des relations d'opinion dans l'entrainement de la classification n'a pas d'impact significatif, l'utilisation des hash tags améliore l'exactitude de la classification d'environ 2%.

Pour confirmer ce résultat, nous avons effectué les mêmes expériences avec un sous-ensemble du corpus initial, dans lesquels ne sont conservés que les tweets contenant des hash tags :

Expériences	EXACTITUDE %
Expérience 1 : sac de mots	79,9
Expérience 2 : hash tag	84,6
Expérience 3 : opinions	80,1
Expérience 4: opinions+hash tags	84,7

Dans ce dernier cas, l'amélioration sur la tâche de classification est passée à 4,8% par rapport à la référence. Ces résultats prouvent sans ambigüité l'impact significatif de l'intégration des hash tags et de leur polarité dans la classification des tweets.

6 Conclusion

Dans ce papier, nous avons proposé un mécanisme de décomposition automatique et d'analyse des « hash tags » de façon à pouvoir utiliser l'information sémantique dont ils sont porteurs dans une tâche de classification. Les différentes expériences que nous avons conduites montrent que ces hash tags ont un impact important dans la polarité des tweets.

De plus, notre méthode présente l'avantage d'être totalement automatique, et permet d'exploiter efficacement près de 80% des « hash tags » complexes présents dans les tweets.

Références

AIT-MOKTHAR, S., CHANOD, J.P. (2002). Robustness beyond Shallowness: Incremental Dependency Parsing. N*umero Special du NLE Journal.*

ASUR, S., & HUBERMAN, B. A. (2010). Predicting the future with social media. Actes de *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference* (Vol. 1, pp. 492-499). IEEE.

AKSHAT BAKLIWAL, PIYUSH ARORA, SENTHIL MADHAPPAN, NIKHIL KAPRE, MUKESH SINGH, AND VASUDEVA VARMA. (2012). Mining sentiments from Tweets. Actes du *3eme Workshop in Computational Approaches to Subjectivity and Sentiment Analysis (WASSA '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 11-18.

BRUN C. (2011). Detecting Opinions Using Deep Syntactic Analysis. Actes de *RANLP, Recent Advances in Natural Language Processing,* Hissar, Bulgaria, September 12-14, 2011.

BRUN C. (2012). Learning Opinionated Patterns for Contextual Opinion Detection, Actes de *Coling2012*, Décembre 2012, Bombay, Inde.

DAVE, K. S., & VARMA, V. (2012, May). Identifying microblogs for targeted contextual advertising. Actes de *Sixth International AAAI Conference on Weblogs and Social Media*.

DAVIDOV D., OREN TSUR, AND ARI RAPPOPORT. (2010). Enhanced sentiment learning using Twitter hashtags and smileys. Actes de *COLING '10*. ACL, Stroudsburg, PA, USA, 241-249.

DEVEAUD, R., & BOUDIN, F. (2012). *LIA/LINA at the INEX 2012 Tweet Contextualization track*. INitiative for the Evaluation of XML Retrieval (INEX).

HU. M., LIU, B. (2004). Mining and summarizing customer reviews. Actes de *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004)*, Seattle, Washington, USA.

JOACHIMS T. (1999). *Making large-Scale SVM Learning Practical.* Advances in Kernel Methods – Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press.

KOEHN, P., & KNIGHT, K. (2003). Empirical methods for compound splitting. Actes de *the tenth conference on European chapter of the Association for Computational Linguistics*-Volume 1 (pp. 187-193). Association for Computational Linguistics.

KOULOUMPIS, E.,WILSON, T.ET MOORE, J. (2011), Twitter Sentiment Analysis: The Good the Bad and the OMG!, in Lada A. Adamic; Ricardo A. Baeza-Yates & Scott Counts, ed., 'ICWSM' , The AAAI Press.

LAKE, T., & FITZGERALD, W. (2011). Twitter Sentiment Analysis. *Western Michigan University, Kalamazoo, MI*, For client William Fitzgerald.

LEVIN, BETH. (1993). English Verb Classes and Alternations A Preliminary Investigation. University of Chicago Press, Chicago and London.

LIU, B. (2010). Sentiment Analysis and Subjectivity, *Chapter of Handbook of Natural Language Processing,* 2nd edition.

MAYNARD D, GREENWOOD, M.A. (2014). Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. *Actes de LREC'2014,* Reykjavik, Finlande.

RAMAGE, D., DUMAIS, S., & LIEBLING, D. (2010). Characterizing microblogs with topic models. Actes de I*nternational AAAI Conference on Weblogs and Social Media* (Vol. 5, No. 4, pp. 130-137).

RUPPENHOFFER, J., ELLSWORTH M, PETRUCK M, JOHNSON C., ET SCHEFFCZYK J. (2005). FrameNet II: Extended theory and practice. *Technical report,* ICSI.

THELWALL, M., BUCKLEY, K.,ET PALTOGLOU, G. (2011). Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology*, 62(2), 406-418.

WANG XIAOLONG, FURU WEI, XIAOHUA LIU, MING ZHOU, ET MING ZHANG. (2011). Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. *Actes de CIKM '11*, Bettina Berendt, Arjen de Vries, Wenfei Fan, Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.). ACM, New York, NY, USA, 1031-1040.

WU, YUANBIN, ZHANG, QI, HUANG, XUANJING, HUANG ET WU, LIDE. (2009). Phrase Dependency Parsing for Opinion Mining. *Actes de EMNLP'09, Vol3.* Association for Computational Linguistics, Stroudsburg PA, USA, 1533-1541.

Modélisation des questions de l'agent pour l'analyse des affects, jugements et appréciations de l'utilisateur dans les interactions humain-agent

Caroline Langlet[1] Chloé Clavel[1]
(1) Institut Mines-Télécom ; Télécom ParisTech ; CNRS LTCI, Paris
caroline.langlet@telecom-paristech.fr, chloe.clavel@telecom-paristech.fr

Résumé. Cet article aborde la question des expressions d'*attitudes* (affects, jugements, appréciations) chez l'utilisateur dans le cadre d'échanges avec un agent virtuel. Il propose une méthode pour l'analyse des réponses à des questions fermées destinée à interroger les attitudes de l'utilisateur. Cette méthode s'appuie sur une formalisation des questions de l'agent – sous la forme d'une fiche linguistique – et sur une analyse de la réponse de l'utilisateur, pour créer un modèle utilisateur. La fiche linguistique de l'agent est structurée par un ensemble d'attributs relatifs, d'une part, à l'attitude à laquelle réfère la question, d'autre part, à sa forme morphosyntaxique. L'analyse de la réponse, quant à elle, repose sur un ensemble de règles sémantiques et syntaxiques définies par une grammaire formelle. A partir des résultats fournis par cette analyse et des informations contenues dans la fiche linguistique de l'agent, des calculs sémantiques sont effectués pour définir la valeur de la réponse et construire le modèle utilisateur.

Abstract. This paper tackles the issue of user's attitudinal expressions in an human-agent interaction. It introduces a method for analysing the user's anwers to the agent's yes/no questions about attitude. In order to build a user model, this method relies on a formalization of the agent's questions and a linguistic analysis of the user's answer. This formalization comprises a set of attributes regarding the attitude expressed and the morphosyntaxic form of the question. The answer is then analyzed by using a set of semantic and syntactic rules included in a formal grammar. Finally, the semantic value of the answer can be calculated by using the results provided by this analysis and the information given by the question formalization. This computation is next integrated in the user model.

Mots-clés : affect, jugement, appréciation, interaction humain-agent, questions-réponses.

Keywords: affect, jugement, appreciation, human-agent interaction, questions-answers.

1 Introduction

Pour améliorer l'interaction avec l'utilisateur, l'un des objectifs clé du domaine des agents conversationnels animés est de doter l'agent d'une capacité de détection des sentiments exprimés par l'utilisateur. Si la plupart des approches proposées prennent en compte le contour prosodique des énoncés et les expressions faciales, le contenu verbal est lui de plus en plus intégré, mais n'est encore que partiellement analysé. Dans le domaine de l'*opinion mining* et du *sentiment analysis*, parallèlement aux méthodes de classification par apprentissage ((Pang & Lee, 2004) et (Turney, 2002), (Riloff & Wiebe, 2003)), se sont également développées des méthodes de détection à base de règles. S'attachant à l'identification des propriétés structurelles de ces expressions (cible, polarité, intensité, etc.), elles reposent sur une analyse logico-sémantique de la structure de la phrase. Ainsi, (Neviarouskaya *et al.*, 2010) et (Moilanen & Pulman, 2007) se basent sur le principe de compositionnalité pour développer des règles de calcul de la polarité. De même, (Shaikh *et al.*, 2009) fournit une adaptation linguistique du modèle OCC (Ortony *et al.*, 1990) à partir de règles fondées sur des indices logico-sémantiques.

Si ces approches offrent l'avantage de fournir une analyse détaillée des expressions d'attitudes/sentiments/opinions, elles ne traitent en revanche que des textes où la composante dialogique est absente. Notre but à long-terme étant de développer un module de détection des attitudes (affects, appréciations, jugements) de l'utilisateur, cette problématique doit être clairement abordée (Clavel *et al.*, 2013). Après avoir proposé dans (Langlet & Clavel, 2014) un modèle d'annotation fondé sur le modèle théorique de (Martin & White, 2005) et permettant d'appréhender les attitudes de l'utilisateur telles qu'elles se manifestent lors de son interaction avec l'agent, le travail présenté ici propose un formalisme pour l'échange de questions-réponses portant sur des attitudes de l'utilisateur. Parmi les formalisations des énoncés interrogatifs déjà pro-

posés, si certains s'attachent à l'aspect logique de leur fonctionnement ((Nelken & Francez, 2002) et (Wisniewski, 2002)), d'autres ont davantage l'objectif de penser ces énoncés relativement à leur fonctionnement en contexte de l'interaction (Ginzburg, 2010). Le formalisme que nous proposons intègre également cette problématique tout en se concentrant sur celle des expressions d'attitudes. Appliqué aux questions de l'agent, il permet de créer une fiche linguistique synthétisant différentes informations tant sur leur contenu sémantique que sur leur forme syntaxique. Cette fiche permet ensuite à un module de traitement des réponses, de créer, après analyse de cette dernière, un modèle de l'utilisateur (Section 2). Après une description générique de ce formalisme, nous présentons comment celui-ci peut être exploité (Section 3) au cours d'un processus d'analyse de réponses à des questions fermées.

2 Un formalisme des questions-réponses portant sur des attitudes

Le traitement des réponses de l'utilisateur à des questions de l'agent l'interrogeant sur ses attitudes repose sur une architecture de gestion de l'interaction impliquant un moteur de dialogue (MD). Suivant les stratégies de dialogue définies, celui-ci décide des énoncés que l'agent doit prononcer au cours de l'intéraction. Pour chacune des questions choisies par le MD, une fiche synthétisant les informations sémantiques qu'elle contient (décrite section 2) est transmise à un module de gestion des réponses (MR). A partir de cette fiche et d'un ensemble de règles (décrites Section 2), le MR procède à l'analyse du contenu verbal de chaque réponse et produit un modèle utilisateur (décrit section 3) qui est envoyé au MD pour alimenter les stratégies de dialogues, à long-terme ou à court-terme, qu'il définit. A long-terme, les informations contenues dans le modèle utilisateur sont utilisées pour définir les préférences de l'utilisateur et planifier les sujets de dialogue abordés par l'agent. A court-terme, elles sont exploitées pour définir des stratégies de réaction et permettent notamment de produire, chez l'agent, des énoncés exprimant un alignement sur le discours de l'utilisateur.

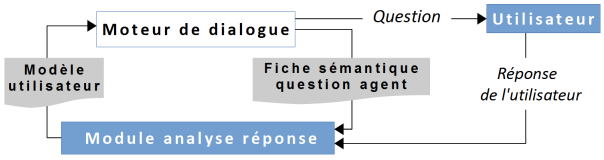


FIGURE 1 – Fonctionnement général du système

Fiche linguistique des questions de l'agent : La formalisation des questions référant à des attitudes a été réalisée à deux niveaux : sémantique et morphosyntaxique.

$AttitudeType = \{appreciation, evaluation\}$ $Source$: prend comme valeur le groupe nominal ou le pronom référant à la source dans l'énoncé interrogatif.	$Polarity = \{negative, postive\}$ $Target$: prend comme valeur le groupe nominal référant à la cible dans l'énoncé interrogatif.

TABLE 1 – Fiche sémantique de l'agent : informations relatives à l'attitude exprimée

$QuestionForm = \{negation, affirmation\}$ renseigne sur le possible caractère interro-négatif de l'énoncé. $Polarity_{AttWord} = \{negative, positive\}$ indique la polarité du mot référant à l'attitude. $Gender_{target} = \{feminine, masculine, neuter\}$ $Number_{target} = \{singular, plural\}$	$Auxiliaire$: forme lemmatisée de l'auxiliaire utilisé pour la formulation de la question. $PosTag_{AttWord} = \{noun, adjective, verb\}$ $AttWord$: forme lemmatisée du mot qui réfère à l'attitude exprimée. $Function_{source} = \{subject, object\}$ & $Function_{target} = \{subject, object\}$ fonction syntaxique de la source et de la cible.

TABLE 2 – Fiche sémantique de l'agent : informations morphosyntaxiques

Une **formalisation sémantique** est en effet nécessaire pour la description de l'attitude à laquelle il est fait référence. Quatre attributs (Table 1), retenus du modèle de (Martin & White, 2005), permettent de décrire l'attitude à laquelle il est fait référence et de définir, en cas de question ouverte, l'objet de la question. Une valeur *undefined* sera ainsi donnée

à l'attribut sur lequel porte la question. Les **informations morphosyntaxiques** (Table 2) permettent quant à elles de détailler la forme de l'énoncé interrogatif pour fournir au MR une représentation la plus fidèle possible.

Sorties – modèle de l'utilisateur : Pour répondre à une question portant sur une attitude l'utilisateur doit soit infirmer ou confirmer l'attitude telle que la décrit la question (réponse à une question fermée), soit préciser la valeur de l'un de ses attributs (réponse à une question ouverte). Formellement, dans le cadre de la méthode d'analyse que nous proposons, le système devra reprendre les attributs tels qu'ils sont définis dans la fiche de l'agent pour les ajouter, avec les valeurs appropriées, au modèle utilisateur. Ce dernier comptera néanmoins un attribut propre, non tiré de la fiche sémantique de l'agent, l'attribut $Modality$, permettant d'informer sur la présence de modalités. Ainsi, comme l'illustre la figure 2, pour une question comme « what's your favorite painter ? », le modèle utilisateur est construit sur la base des attributs contenus dans la fiche agent – uniquement dans sa partie sémantique – pour lesquels une analyse de la réponse aura permis de décider que la valeur de l'attribut $Target$ est $Picasso$ et que la valeur des autres attributs peut être conservée.

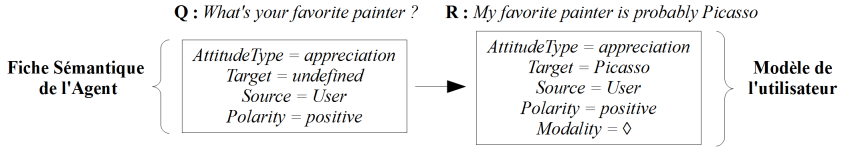


FIGURE 2 – Création du modèle utilisateur

3 Système d'analyse : le cas des questions fermées

3.1 Caractérisation des questions fermées et de leurs réponses

Lors d'un précédent travail, décrit dans (Langlet & Clavel, 2014), 15 transcriptions du corpus Semaine (McKeown *et al.*, 2011) ont été annotées manuellement [1]. Parallèlement à l'annotation des attitudes de l'utilisateur, les énoncés de l'agent ont été annotés relativement à l'acte illocutoire qu'ils accomplissent [2]. Pour chaque actes illocutoires annotés, une structure de traits permet de spécifier si l'énoncé réfère à une attitude. Lorsque cela est le cas, deux unités sont utilisées pour annoter la source et la cible impliquées. Pour typer les questions fermées et leurs réponses, ont été sélectionnés, d'une part, tous les énoncés de l'agent annotés comme *directifs* (109 au total) et référant à une attitude dont la source est définie comme étant l'utilisateur, et, d'autre part, tous les énoncés de l'utilisateur succédant immédiatement à ces énoncés – et pouvant donc être considérés comme de potentielles réponses. Dans cette première sélection ont été comptés 38 questions ouvertes, 59 questions fermées et 20 énoncés exprimant un conseil ou une suggestion. L'extraction des questions fermées s'est ensuite faite manuellement.

Fonctionnement sémantique des réponses aux questions fermées : A la différence des questions ouvertes, qui interrogent sur un objet précis dont la définition est laissée à l'initiative de l'utilisateur, les questions fermées se présentent comme des propositions ($?p$) que les réponses qu'elles sollicitent devront valider ou invalider. Ainsi la question fermée « Do you like outdoors activities ? » engage l'utilisateur à valider ou à invalider la proposition « you like outdoors activities ». Pour répondre à une question fermée, l'utilisateur a ainsi deux possibilités : (i) soit exprimer ***une confirmation ou une infirmation simple***, la première définissant le contenu propositionnel de la question comme vrai, la seconde comme faux ; (ii) soit exprimer ***une confirmation ou une infirmation modalisée***, faisant porter sur le contenu propositionnel une modalité pouvant être aléthique, déontique, temporelle ou épistémique (Le Querler, 1996). 15 réponses de ce type – sur un total de 59 – ont pu être rencontrées dans le sous-corpus d'étude. Pour résumer de manière formelle le fonctionnement sémantique de ces deux types de réponse, il est possible de dire qu'elles définissent une valeur pour un attribut $Value_p$ prise dans l'ensemble $\{p, \neg p, \diamond p, \Box p, \diamond \neg p, \Box \neg p\}$. Lorsque la réponse fait suite à une question fermée référant à une attitude, la valeur qu'elle attribue à $Value_P$ va également entrainer une confirmation ou une infirmation de la valeur que la question de l'agent avait attribuée à $Polarity$. En cas d'une infirmation, la valeur de l'attribut sera ainsi inversée. Il est également important de noter que de même qu'une confirmation ne s'exprime pas nécessairement par un *yes* (ou ces synonymes), une infirmation ne l'est pas nécessairement par un *no* (ou ces synonymes). Ainsi, dans l'énoncé interro-négatif « Don't you love outdoors activities », le contenu propositionnel « you don't love outdoors activities » est affirmé par la réponse

1. Le corpus Semaine compte au total 65 transcriptions manuelles de sessions où un utilisateur humain interagit avec un operator humain jouant le rôle d'un agent virtuel
2. Nous nous référons ici à la classification définie dans (Searle, 1976) et retenant cinq actes : les acte directifs, commissifs, assertifs, expressifs ou de déclaration.

« no » et infirmée par « yes ». Afin de pouvoir déterminer la valeur des l'attribut $Value_p$ et par la suite celle de l'attribut $Polarity$, un typage de la réponse permettant savoir si elle équivaut à un *yes* ou à un *no* est nécessaire.

Forme morpho-syntaxique des réponses aux questions fermées : Sur le plan morpho-syntaxique, la réponse de l'utilisateur devra donc nécessairement comporter une séquence permettant d'infirmer ou de confirmer, simplement ou avec modalité, le contenu propositionnel de la question. Cependant, les réponses de l'utilisateur peuvent ne pas se limiter à cette séquence dédiée à la confirmation ou à l'infirmation (modalisée ou non). L'utilisateur peut être amené à formuler une réponse plus longue apportant des informations supplémentaires (dans le corpus d'étude 28 réponses – sur 51 – se limitent à l'expression d'une séquence exprimant une infirmation ou une confirmation). Parmi ces informations, l'utilisateur peut être amené à développer des attitudes additionnelles. Si elles sont minoritaires dans le sous-corpus d'étude (8 cas relevés sur 59), il est important de pouvoir les traiter quand elles apparaissent et d'intégrer les résultats de leurs analyses dans le modèle utilisateur (Section 3.3).

3.2 Typage de la réponse : *yes*, *no* ou *réponse modale* ?

Détection des séquences *yes* et *no* Pour définir si une réponse exprime un *yes* ou un *no*, il est nécessaire de ne pas se contenter du seul repérage de ces deux lexies dans les réponses de l'utilisateur. D'une part, en plus de leurs variantes oralisées (*yeah*, *yep*, *nope*), des adverbes synonymes peuvent être employés (*indeed*, *of course*, etc). D'autre part, ces différentes unités lexicales peuvent être employées conjointement à des adverbes d'intensité comme *very*, *extremely*, etc. La grammaire intègre donc la règle suivante : $answer(yes|no) \rightarrow (yesWd|noWd)+, advIntens?$.

Des formulations plus complexes peuvent également apparaître : pour exprimer un *yes* ou un *no*, l'utilisateur peut procéder à une reprise de l'auxiliaire ou du noyau verbal de la question. Cette reprise peut-être employée seule ou accompagnée d'un $yesWd$ ou d'un $noWd$ (par exemple : « Does they make you happy » → « yes, they do»). Pour gérer ce phénomène de reprise, la grammaire prend en compte les valeurs de l'ensemble des attributs morphosyntaxiques de la fiche de l'agent, notamment celles des attributs $Function_{source}$ et $Function_{target}$. Les règles prennent ainsi la forme décrite dans la figure 3. Le non-terminal $Target$ a comme terminal (i) le littéral fourni par l'attribut $Target$ dans la fiche agent, (ii) un pronom personnel aligné en genre et en nombre sur les valeurs fournies par les attributs $Gender_{target}$ et$Number_{target}$ dans la fiche agent. Le non-terminal aux prend comme terminal la forme lemmatisée de l'auxiliaire donné par l'attribut $Auxiliaire$.

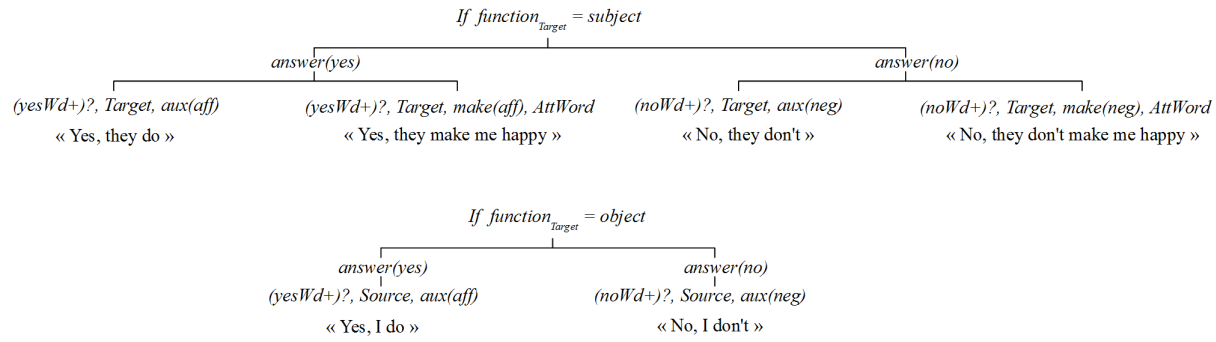


FIGURE 3 – Règles pour l'analyse des réponses avec reprise

***yes*|*no* modalisé** Pour la détection d'unités lexicales exprimant des modalités, un court lexique a été constitué. Il compte pour le moment une vingtaine d'entrées, mais sera par la suite augmenté. Pour chaque entrée du lexique, sont spécifiées sa valeur modale et sa catégorie grammaticale. A la manière des réponses de type *yes*|*no*, ces réponses peuvent être limitées dans leur forme syntaxique à l'emploi d'unités lexicales ou de courtes séquences syntaxiques exprimant uniquement une modalité. Par exemple, à la question « Will you enjoy holidays ? », l'utilisateur peut être amené à formuler l'énoncé « maybe yes », exprimant la probabilité (modalité épistémique). Selon la catégorie grammaticale de la lexie exprimant une modalité trois types de règles sont définies par la grammaire (cf tableau 3). Notons qu'une valeur modale peut porter sur un adverbe de négation (ex : « maybe not ») et que, inversement, un négation peut porter sur une unité exprimant une négation. Dans ces deux cas, la valeur modale de la réponse sera donc différente de celle de la lexie utilisée. Lorsque la modalité porte sur la négation, la valeur modale de la réponse sera contraire ou sub-contraire de celle de la lexie (le possibilité, $\diamond$, devient la contigence, $\diamond\neg$). En revanche lorsque la négation porte sur la modalité, la valeur modale de la réponse est alors la modalité contradictoire de celle de la lexie (la possibilité, $\diamond$, devient l'impossibilité, devient $\neg\diamond$).

$answer(valMod:X) \rightarrow [(yesWd+)|(noWd+)]?, modWd(cat:adv, valMod:X)$
$answer(valMod:X) \rightarrow [(yesWd+)|(noWd+)]?, it, is, modWd(cat:adj, valMod:X)$
$answer(valMod:X) \rightarrow [(yesWd+)|(noWd+)]?, Source, modWd(cat:vb, valMod:X)$

TABLE 3 – Règles pour la détection des réponses avec valeur modale

Ce type de réponse peut également intégrer, comme pour les réponses de type *yes|no*, des phénomènes de reprise. Les patrons syntaxiques sont alors les mêmes que ceux décrits dans le paragraphe précédent, à l'exception d'une insertion d'un adverbe modal soit devant la reprise verbale, soit devant la séquence $Source, aux$.

3.3 Caractérisation des attributs $Value_P$ et $Polarity$

Calcul de $Value_P$: Une fois déterminé le type de la réponse (*yes*, *no* ou *réponse modale*), le système lance le calcul de la valeur de l'attribut $Value_P$. Ce calcul n'est effectué que pour les réponses de type *yes|no*. En effet, pour les réponses modales, il n'est nécessaire que de récupérer la valeur modale identifier par la grammaire et de l'attribuer à p. Pour calculer cette valeur, les règles présentées ci-dessus s'appuient sur le type de la réponse et la valeur de l'attribut $QuestionForme$ (fiche de l'agent). Ainsi, lorsque la réponse équivaut à un *yes* : si $QuestionForm = affirmation$ alors $value_P = p$, si $QuestionForm = negation$ alors $value_P = \neg p$. En revanche, lorsque la réponse équivaut à un *no*, si $QuestionForm = affirmation$ alors $value_P = \neg p$, si $QuestionForm = negation$ alors $value_P = p$.

Calcul de $Polarity$: A partir de la valeur attribuée à $Value_P$, la valeur de $Polarity$ peut être définie. Lorsque p est nié, l'attribut est $Polarity$ prend la valeur inverse de celle de la fiche de l'agent : $if Value_P = \neg p, alors : Polarity_{user} = reverse(Polarity_{agent})$. La valeur sera en revanche conservée lorsque p est défini comme vrai par la réponse de l'utilisateur : $if Value_P = p, alors Polarity_{user} = Polarity_{agent}$. Pour les *yes|no* modalisés l'inversion de la polarité n'est appliquée que pour les valeurs modales de l'ordre contigent/constestable ($\diamond\neg$) et de l'impossible/exclu ($\neg\diamond$).

Analyser les expressions d'attitude additionnelles Pour mener à bien cette analyse, deux types de ressources sont utilisés : une ressource lexicale, *Sentiwordnet* (Andrea & Fabrizio, 2006) et une grammaire formelle analysant le contexte syntaxique des lexies référant à une attitude. Lorsque, dans la réponse de l'utilisateur, une lexie a été identifiée comme appartenant à Sentiwordnet (non-terminal $LexieAtt$ dans la grammaire), le système lance la grammaire afin de pouvoir en définir le type (affect, évaluation [3]), la cible et de vérifier que l'utilisateur en est bien la source. Les règles définies par la grammaire s'appuient sur les patrons syntaxiques décrits dans le tableau 3.3. Le type de l'attitude y est défini relativement à la nature et la fonction de la lexie attitude, mais aussi à la place occupé par la source, ici toujours un pronom personnel de première personne (les attitudes que nous cherchons à repérer sont celles dont l'utilisateur est la source). Les $Target$ prennent dans ces patrons soit la forme de syntagmes nominaux, soit la forme de propositions infinitives. La classe $vbOpinion$ regroupe des verbes signifiant que le sujet énonce une opinion (*find*, *believe*, *think*). La polarité est calculée à partir de la polarité attribuée à la lexie dans Sentiwordnet. Sous les effets d'une négation, cette polarité peut s'inverser lorsqu'une négation porte sur la lexie attitude.

$eval \rightarrow I, vbOpinion, Target, be?, det?, adjAttitude, noun?$	« I find this picture beautiful »
$eval \rightarrow Target, vbAttributif, [(det, AdjAttitude, noun)\|AdjAttitude]$	« This picture looks beautiful »
$eval \rightarrow I, vbAttitude, Target$	« I love this picture »
$affect \rightarrow I, have, det?, adjAttitude, noun$	« I have »
$affect \rightarrow I, [vbAttributif\|feel], adjAttitude$	« I feel miserable »
$affect \rightarrow I, vbAttitude$	« I was laughing »
$affect \rightarrow Target, make, me, AdjAttitude$	« He makes me sad »
$affect \rightarrow I, be, confronted, to, NounAttitude$	« I was contronted to a problem »
$affect \rightarrow Target, vbAttitude, me$	« He frustrates me »

3. Pour l'analyse des attitudes additionnelles, les deux catégories, *appréciation* et *jugement*, ont été regroupées au sein d'une même catégorie *evaluation*. En effet, ces deux catégories ont, sur le plan de l'expression, un certain nombre de patrons en commun. De plus, leur distinction repose essentiellement sur la nature ontologique de leur cible (artefacts et processus, pour les appréciations, comportements et personnes, pour les jugements). Or, cette distinction ne peut être fait en l'état actuel de nos travaux.

4 Conclusion et perspectives

Dans cet article, nous proposons une méthode permettant, dans le cadre d'échange de questions-réponses entre un agent virtuel et un utilisateur humain, de produire un modèle de l'utilisateur à partir d'une analyse linguistique de la réponse et de l'exploitation d'une formalisation de la question de l'agent. Cette formalisation concerne tant des informations morphosyntaxiques que des informations sémantiques relatives à l'attitude à laquelle réfère la question. Chacune d'elles peuvent ensuite être utilisées au cours de l'analyse de réponse : soit pour en reconnaître la forme syntaxique, soit pour en faire l'interprétation sémantique. Cette méthode offre l'avantage de prendre en compte la diversité formelle des réponses à des questions fermées (prise en compte de valeurs modales et d'attitudes additionnelles) pour mieux détailler le modèle utilisateur. Si la méthode est ici présentée pour l'analyse de réponses à des questions fermées, le formalisme sur lequel elle s'appuie a été conçu pour être fonctionnel également avec les questions ouvertes. Des travaux ultérieurs auront donc comme objectif de développer des règles pour un système d'analyse des réponses à des questions ouvertes ainsi que d'intégrer dans la fiche linguistique de l'agent les informations véhiculées par le contenu non-verbal. Au final, une évaluation finale sur un autre échantillon plus large issu du corpus Semaine permettra de vérifier la fiabilité de la méthode.

Références

ANDREA E. & FABRIZIO S. (2006). Sentiwordnet : A publicly available lexical resource for opinion mining. In *5th Conference on Language Resources and Evaluation*, p. 417–422.

CLAVEL C., PELACHAUD C. & OCHS M. (2013). User's sentiment analysis in face-to-face human-agent interactions – prospects. In *Workshop on Affective Social Signal Computing, Satellite of Interspeech* : Association for Computational Linguistics.

GINZBURG J. (2010). *Questions : Logic and Interaction*, In J. VAN BENTHEM & A. TER MEULEN, Eds., *Handbook of Logic and Language*.

LANGLET C. & CLAVEL C. (2014). Modelling user's attitudinal reactions to the agent utterances : focus on the verbal content. In *5th International Workshop on Corpora for Research on Emotion, Sentiment & Social Signals (ES3 2014)*, Reykjavik, Iceland.

LE QUERLER N. (1996). *Typologie des modalités*. Presse Universitaire de Caen.

MARTIN J. R. & WHITE P. R. (2005). *The Language of Evaluation. Appraisal in English*. London and New York : Macmillan Basingstoke.

MCKEOWN G., VALSTAR M., COWIE R., PANTIC M. & SCHRODER M. (2011). The semaine database : Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, **3**(1), 5–17.

MOILANEN K. & PULMAN S. (2007). Sentiment composition. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, p. 378–382.

NELKEN R. & FRANCEZ N. (2002). Bilattices and the semantics of natural language questions. *Linguistics and Philosophy*, **25**, 37–64.

NEVIAROUSKAYA A., PRENDINGER H. & ISHIZUKA M. (2010). Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, p. 806–814, Stroudsburg, PA, USA : Association for Computational Linguistics.

ORTONY A., CLORE G. & COLLINS A. (1990). *The Cognitive Structure of Emotions*. Cambridge, University Press.

PANG B. & LEE L. (2004). A sentiment education : Sentiment analysis using subjectivity summarization based on minimum cut. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, p. 486–497, Stroudsburg, PA, USA : Association for Computational Linguistics.

RILOFF E. & WIEBE J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, p. 105–112.

SEARLE J. R. (1976). A classification of illocutionary acts. *Language in society*, **5**(01), 1–23.

SHAIKH M., PRENDINGER H. & M. I. (2009). A linguistic interpretation of the occ emotion model for affect sensing from text. In *Affective Information Processing*, p. 378–382 : Springer London.

TURNEY P. (2002). Learning extraction patterns for subjective expressions. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, p. 417–424.

WISNIEWSKI A. (2002). Question and inferences. *Linguistics and Philosophy*, **25**, 37–64.

KNG: un outil pour l'écriture facile de cascades de transducteurs

François Barthélemy[1, 2]
(1) INRIA – Alpage, domaine de Voluceau 78153 Rocquencourt
(2) CNAM – Cedric, 292 rue Saint-Martin, 75003 Paris
francois.barthelemy@inria.fr

Résumé. Cet article présente une bibliothèque python appelée KNG permettant d'écrire facilement des automates et transducteurs finis. Grâce à une gestion soigneuse des codages et des entrées-sorties, cette bibliothèque permet de réaliser une cascade de transducteurs au moyen de tubes unix reliant des scripts python.

Abstract. This paper presents a Python library called KNG which provides facilities to write easily finite state automata and transducers. Through careful management of encodings and input/output, a transducer cascade may be implented using unix pipes connecting python scripts.

Mots-clés : Machines finies à états, cascade de transducteur, expression régulière.

Keywords: Finite State Machines, Transducer Cascade, Regular Expression.

1 Chaînes de traitements et transducteurs finis

Les machines finies à états, c'est à dire les automates et transducteurs finis, ont conquis des domaines d'application pour lesquels ils sont bien adaptés du fait de leur efficacité. Dans le domaine du TAL, il s'agit entre autres de la représentation de lexiques, de l'analyse morpho-lexicale, de l'étiquetage syntaxique, de diverses formes de recherche de motifs ainsi que de l'approximation de tâches pour lesquels elles ne sont pas assez puissantes comme l'analyse syntaxique.

D'une part des outils ont vu le jour pour faciliter la description de machines finies : ce sont les expressions régulières et leurs extensions, notamment les règles de réécriture contextuelles. Il faut noter d'ailleurs que ce type de règles est utilisé spécifiquement pour le TAL. D'autres part, les expressions régulières sont devenues un mécanisme de traitement des chaînes de caractères important dans de nombreux langages de programmation. Mais les deux mondes sont restés jusqu'à ce jour relativement distincts : les systèmes dédiés aux machines finies ne communiquent pas facilement avec d'autres applications alors que les langages comme perl ne proposent pas d'opérations combinant deux machines finies ou ayant une machine finie comme résultat.

Pour bien comprendre la différence, prenons l'exemple de la substitution de sous-chaînes. Il est facile d'écrire un script Perl de 5 lignes qui prend des chaînes de caractères et remplace dans ces chaînes toutes les occurrences d'une sous-chaîne s1 par une autre sous-chaîne s2. Un transducteur fini peut faire le même travail sur les chaînes, mais il peut aussi s'appliquer de façon plus générale à n'importe quel ensemble régulier de chaînes. On peut appliquer un transducteur fini à un langage régulier et obtenir comme résultat un autre ensemble régulier. C'est plus efficace que de procéder chaîne par chaîne : d'une part cela permet de traiter en temps fini un ensemble infini de chaîne, s'il est régulier. D'autre part cela permet un gain de complexité grâce au partage des calculs : on passe d'une complexité exponentielle à une complexité quadratique.

Plus spécifiquement pour le TAL, l'utilisation de machines finies, éventuellement avec une restriction aux machines acycliques, permet de représenter l'ambiguïté de certains composants et de considérer en parallèle plusieurs analyses alternatives. Cette aptitude est utilisée par certaines techniques de correction d'erreurs où plusieurs corrections peuvent être proposées.

L'utilisation de poids dans des machines finies pondérées permet de gérer l'ambiguïté : à certains stades, l'ambiguïté peut être résolue ou diminuée au moyen d'un algorithme de détermination du meilleur ou des n meilleurs chemins.

Dans cet article, nous décrivons une bibliothèque python permettant d'écrire facilement des machines finies pondérées avec un haut degré d'intégration au langage hôte. Cette bibliothèque permet non seulement de créer des automates et transducteurs, mais elle permet également de les importer et exporter facilement sous différents formats. Cela permet la

réalisation de cascades de transducteurs sous la forme de tubes unix (pipes) de scripts pythons.

Les principales caractéristiques de la bibliothèque seront exposées : la représentation des machines ; les opérations ; la représentation des caractères et leur codage ; les entrées-sorties ; la compilation statique dans un langage interprété. Nous terminerons avec une rapide comparaison avec d'autres langages de machines finies.

2 KNG et les machines finies

KNG est un module python qui offre un certain nombre de fonctions et de classes permettant la représentation et la manipulation de machines finies pondérées. En premier lieu, il y a deux classes différentes pour les automates (KNG.automaton) et les transducteurs (KNG.transducer). Une machine finie pour KNG est un objet instance de l'une de ces deux classes. Ces objets peuvent être stockés dans des variables python, passés en paramètre à des fonctions et des méthodes. Ce sont donc des valeurs python au plein sens du terme.

La construction des objets peut se faire de deux façon différentes : directement par le constructeur de la classe, ou de façon incrémentale en ajoutant successivement des éléments au moyen de méthodes. La construction directe se fait au moyen d'une expression régulière représentée par une chaîne de caractère python. Pour la construction incrémentale, une forme mutable de machine initialement vide est créée par le constructeur, puis des éléments sont ajoutés par des appels de méthodes. Ces éléments sont soit des noeuds et arcs, soit des chaînes ou paires de chaînes appartenant à l'ensemble régulier que la machine représente.

```
# construction directe (alp : alphabet défini préalablement)
ex1=KNG.automaton(alp,'(a|b|c)(d|e)')
# construction incrémentale 1
ex2==KNG.automaton(alp)
ex2.add_path('ad')
ex2.add_path('ae')
# construction incrémentale 2
ex3==KNG.automaton(alp)
st1=ex3.add_state()
st2=ex3.add_state()
ex3.add_arc(st1,st2,'a')
ex3.set_initial(st1)
ex3.set_final(st2)
```

Pour la représentation interne des machines, KNG utilise la bibliothèque OpenFst de google, qui est un outil écrit en C++. L'accès à cette bibliothèque est transparent pour l'utilisateur qui ne voit que le module python KNG. Le type de poids utilisé est le type standard des poids d'OpenFst, à savoir le semi-anneau tropical.

Les opérations sur automates et transducteurs sont des méthodes des classes considérées. Dans les deux classes, il y a l'union, la concaténation, la clôture étoile, la clôture plus et l'opération qui extrait les n chemins de poids minimal. Les opérations binaires (union et concaténation) ne sont définies que pour deux objets de la même classe, à savoir deux automates ou deux transducteurs. Les opérations spécifiques des automates sont l'intersection, la complémentation et la différence ensembliste. Les opérations spécifiques aux transducteurs sont la composition, l'inversion et l'application fonctionnelle (réécriture d'une entrée en une sortie).

```
# operation destructive, resultat dans ex1
ex1.concat(ex2)
# ex1 inchangé, résultat dans ex4
ex4=ex1.concat(ex2,'conservative')
# chaîne d'applications
ex1.concat(ex2).union(ex3).rm_epsilon()
```

Il y a ensuite des opérations de conversion d'une classe dans l'autre : le produit cartésien construit un transducteur à partir de deux automates. L'identité construit un transducteur à partir d'un seul automate. Les deux projections extraient l'un

ou l'autre côté d'un transducteur, ce qui donne un automate. La dernière opération de conversion est la plus complexe : c'est la *règle de réécriture contextuelle* (Kaplan & Kay, 1994). Elle prend quatre opérandes qui sont quatre automates finis représentant respectivement un motif à remplacer, un motif de remplacement, le contexte gauche et le contexte droit. Le résultat de l'opération est un transducteur fini qui réécrit en une seule fois toutes les occurrences du motif respectant les conditions de contexte.

Il existe trois opérations d'optimisation d'une machine finie : l'élimination des epsilon-transitions, la déterminisation et la minimisation.

KNG offre des mécanismes de conversion vers les chaînes de caractères de python. Enumérer les chaînes reconnues par un automate se fait au moyen d'un itérateur ce qui permet l'utilisation de la boucle for. Par exemple le code suivant permet d'afficher les chaînes d'un automate.

```
auto=KNG.automaton("(a|b|c)(d|e)")
for st in auto:
    print st
```

Si le langage de l'automate est infini, la boucle ne termine jamais. Dans l'autre sens, la conversion d'une séquence de chaînes en un automate se fait au moyen de la construction incrémentale de l'objet. Pour les transducteurs, ce ne sont pas des chaînes mais des paires de chaînes qui sont énumérées (type `tuple` de python).

3 Alphabets, codages, entrées/sorties

Par définition, les machines finies sont définies sur un ensemble fini de symboles appelé un alphabet. Les éléments de cet ensemble peuvent être des caractères, c'est à dire des composants de base des chaînes de python, mais ils peuvent également être d'une autre nature. Les caractères peuvent être représentés en machine selon différents codages : iso-8859, unicode, utf-8, etc.

Un exemple de symboles d'ensemble finis qui ne sont pas des caractères et qui sont couramment utilisés en linguistique dans des machines finies sont les étiquettes syntaxiques telles que Prep, Det, Adv, etc. En KNG, ces symboles sont identifiés au moyen d'un identificateur qui commence par une lettre et se poursuit par une suite quelconque de lettres, de chiffres et de soulignés. Dans la suite nous les appellerons *symboles multicaractères*.

En interne, les machines finies utilisent une représentation des caractères sous forme de nombres entiers. Les caractères peuvent être représentés en utilisant un codage au choix parmi quelques codages prédéfinis et l'utilisateur peut également spécifier sa propre table de conversion. Pour ce qui est des symboles multicaractères, KNG leur octroie un code dans les plans privés d'unicode sauf si l'utilisateur spécifie lui-même le codage à employer. Dans les expressions régulières, les symboles multicaractères sont entourés de < et > (par exemple `<Prep>`). Un alphabet spécifie le codage utilisé en interne dans la bibliothèque. Il n'exclut pas l'utilisation d'autres codages pour les entrées-sorties.

Les alphabets de KNG sont structurés en classes de symboles qui sont simplement des ensembles finis, pas nécéssairement disjoints, de symboles. Ces classes ont des noms qui peuvent être utilisés dans les expressions régulières pour dénoter la disjonction de leurs éléments. Les alphabets sont compilés en objets instances de la classe `KNG.alphabet`. Les constructeur des classes KNG.automaton et KNG.transducer prennent un objet alphabet en paramètre. Les opérations impliquant plusieurs machines ne sont définies que pour les machines utilisant le même alphabet.

```
alpstr = '''
using unicode;
class lettre="a..zéèêàùçôûâïëüÿ";
class sep=" -";
class chiffre="0123456789";
class trait="<masc><fem><sg><pl><ord><card>";
'''
alp = KNG.alphabet(alpstr)
```

KNG permet de faire des entrées-sorties de machines finies et d'alphabets au moyen de la notion de flot (stream). Cette

construction présente dans les langages de programmation modernes permet un abord abstrait de différent types d'entrées-sorties : entrée et sortie standards, fichiers et périphériques divers.

Un flot est une séquence de valeurs de même type. KNG propose des flots de machines finies et des flots d'alphabets. Ces flots peuvent utiliser différentes représentations textuelles ou binaires. Les flots binaires actuellement implémentés sont deux variantes du format binaire OpenFst : ce format sans modification, adapté pour communiquer avec les commandes OpenFst ; ce format enrichi avec des informations à propos de l'alphabet utilisé, adapté pour la communication entre scripts KNG. Les formes binaires, comme on peut s'y attendre, sont plus efficaces mais moins portables que les représentations textuelles.

Les formes textuelles sont plus adaptées pour communiquer avec des outils externes ou des êtres humains. Il y a d'abord les expressions régulières, limitées pour l'instant aux flots d'entrées. Un tel flot est une séquence d'expressions régulières. Chacune d'elle est compilée par l'opération de lecture en un objet KNG.automaton ou KNG.transducer. Nous souhaitons à l'avenir offrir également les expressions régulières pour les flots de sortie. Une autre forme textuelle acceptée par KNG est le format FSM qui décrit les états et les arcs de la machine. Ce format créé à l'origine pour FSM (Mohri *et al.*, 2002) est accepté par différents outils, notamment Openfst et Xfst. Nous voudrions ajouter ultérieurement des formats XML de représentation de machines finies tels que FSM-XML (Demaille *et al.*, 2009) ou SCXML (W3C, 2012).

Les formats textuels sont acceptés sous différents codages (utf-8 et quelques codages de la norme iso-8859) et ce de façon indépendante du codage utilisé en mémoire pour les objets créés. Par exemple, KNG peut lire un flot codé en iso-8859-1 pour créer un objet utilisant le codage unicode.

Les flots dans KNG sont implémentés par des classes. Généralement, l'utilisateur ne crée pas directement des objets de ces classes mais utilise une fonction `open` qui construit et renvoie un objet d'une classe ou d'une autre en fonction de la valeur des paramètres. KNG s'inspire ici de la gestion des flots par python. La fin de flot se traduit par la levée d'une exception. Les flots en écriture offrent deux méthodes : l'écriture d'une valeur (write) et la fermeture du flot (close). Les flots de lecture offrent une méthode de lecture (read) qui renvoie une machine finie ou un alphabet et des itérateurs permettant d'utiliser directement un flot dans une instruction `for`.

```
# flot d'entrée binaire dans un fichier, codage utf-8
input_stream_1=open('converter.bin','r','machine','utf-8',alp)
trans=input_stream_1.read()
# flot d'entrée textuel expression régulière, entrée standard
input_stream_2=KNG.open(sys.stdin,'r','machine','utf-8','regexp',alp)
# flot de sortie textuel fsm, sortie standard
output_stream=KNG.open(sys.stdout,'w','auto','utf-8','fsm')
# iterateur sur input_stream_2, aut est une machine finie
for aut in input_stream_2:
   output_stream.write(trans.apply(aut,'lr'))
output_stream.close()
```

La méthode `apply` utilisée dans la boucle est l'application fonctionnelle du transducteur. A chaque tour de boucle, une expression régulière lue sur l'entrée standard est compilée en une machine finie désignée par le nom `aut`, utilisée comme entrée pour le transducteur `trans`, et la sortie du transducteur est écrite sur la sortie standard au format `fsm`.

4 Compilation et langage interprété

Python est un langage interprété. Cela signifie qu'une expression régulière apparaissant dans un programme est recompilée à chaque exécution du programme, même si elle ne change pas d'une exécution à l'autre. Ce n'est clairement pas ce que désire une personne écrivant une cascade de transducteur : elle souhaite que tout le travail faisable statiquement ne soit fait qu'une seule fois. Sur ce point, KNG s'est inspiré de ce que fait PLY, une implémentation en python de LEX et YACC. Une grammaire PLY est un programme python. A la première exécution de ce programme, PLY compile la grammaire et crée des tables sauvegardées dans des fichiers. Aux exécutions suivantes, PLY vérifie que la grammaire est inchangée. Si c'est le cas, les fichiers sont relus. Sinon, la grammaire est recompilée.

KNG procède de façon analogue. Par une analyse de la façon dont une machine est créée, KNG détecte si c'est une construction statique ou dynamique. Par exemple, une expression régulière donnée sous forme d'une chaîne de caractère

constante est statique. Une expression régulière lue sur un flot donne naissance à un objet dynamique. En cas de doute, une machine est considérée comme dynamique.

KNG sauvegarde dans des fichiers les machines statiques. Lors d'une nouvelle exécution du programme, les dates du fichier source python et de la machine sauvegardée sont comparées pour déterminer si la description de la machine a pu changer entre temps. Si ce n'est pas le cas, le fichier de sauvegarde est relu. Par ailleurs, dans ce processus de sauvegarde des objets statiques, les dépendances entre machines sont représentées et sauvegardées. Il se peut que la définition d'une machine soit inchangée mais qu'un élément précurseur ait évolué rendant nécessaire un recalcul. Par exemple, si une machine A est obtenue en réalisant une clôture étoile d'une machine B, tout changement de B oblige à recalculer A. C'est pour cela qu'une relation de dépendance entre composants doit être maintenue et mémorisée.

Dans son état actuel, la gestion des dépendances par KNG est assez rudimentaire : les dépendances sont enregistrées entre fichiers sur la base de la date de dernière modification. La détermination des éléments statiques est également rudimentaire. Des sous-composants statiques de machines dynamiques pourraient être sauvegardés.

5 Relations entre KNG et Python

KNG n'utilise pas d'espace de noms spécifique. Pour nommer une machine, on peut utiliser tout simplement une variable python qui contiendra une référence à l'objet implémentant la machine. C'est dans ce cas l'espace de noms du programme utilisateur qui est utilisé. Une autre solution est de stocker la paire (nom,valeur) dans un dictionnaire (tableau associatif) Python. Ce qui est moins simple et moins courant, c'est que ce nom peut être utilisé à l'intérieur d'une expression régulière. Pour distinguer le nom d'une machine préexistante `m` d'une séquence de caractères, le nom doit être encadrés par deux symboles ‘ (bacquote) : `` `m` `` est un nom, `m` le caractère m.

La variable n'appartient de toute façon pas à l'espace de nom du module KNG qui n'est ni celui du programme utilisateur, ni le dictionnaire utilisé pour stocker les machines finies. Si une expression régulière contient des références à des machines préexistantes, l'espace de nom à utiliser doit être passé en paramètre au compilateur KNG.

```
auto1=KNG.automaton(alp,'cd')
# auto1 variable python
# espace de nom: celui du script python
auto3=KNG.automaton(alp,'ab|`auto1`|ef',globals())
# le troisième paramètre est l'espace de nom
```

L'utilisation de l'espace de noms du programme utilisateur permet de structurer les définitions de machines finies au moyen de paquetages Python. Par exemple, si l'on définit une machine `m1` dans un paquetage `A`, on pourra la référencer dans une expression régulière d'un programme qui importe `A` au moyen du nom qualifié `A.m1`. On peut utiliser les mécanismes de visibilité Python (à savoir la variable __all__ du fichier __init__.py) pour spécifier si une machine est visible ou non à l'extérieur d'un paquetage.

Les constructions de programmation Python (boucles, exceptions, classes, fonctions) peuvent être utilisées pour une application en KNG. Par exemple, une fonction peut être définie qui prend en paramètre un automate fini, réalise l'application fonctionnelle de plusieurs transducteurs en cascade, puis optimise le résultat de cette cascade au moyen d'élimination d'epsilon-transitions, déterminisation et minimisation et enfin renvoie l'automate résultant de cette séquence.

Comme nous l'avons déjà mentionné, KNG procure des itérateurs permettant de parcourir des ensembles de valeurs au moyen d'un boucle for. Ces itérateurs sont disponibles sur les machines (itération sur les chaînes du langage d'un automate, itération sur les paires de la relation d'un transducteur) et sur les flots d'entrées (itération sur les alphabets ou les machines selon le type du flux).

6 Comparaison avec d'autres outils

Il est impossible de faire une comparaison exhaustive avec tous les langages et systèmes existants qui sont fort nombreux. Nous allons prendre quelques exemples représentatifs des systèmes utilisés pour le TAL.

Commençons par la façon dont Perl utilise les expressions régulières. Les expressions régulières de Perl sont utilisées pour spécifier deux opérateurs et une fonction de traitement de chaînes : la reconnaissance de motif, la substitution et le découpage respectivement notés `m//`, `s///` et `split`. Il n'y a pas véritablement d'opérateur permettant de combiner des expressions régulières. Tout au plus une expression régulière quotée peut-elle être utilisée comme constituant d'une autre expression régulière. Les opérateurs de reconnaissance et de substitution ne s'appliquent qu'à des chaînes et leur résultat est une chaîne.

L'opérateur de substitution n'est pas un transducteur fini : en effet la chaîne qui remplace la sous-chaîne appariée au motif de la substitution est calculée dynamiquement, une fois liées les variables du motif. Le calcul peut d'ailleurs inclure l'évaluation d'une expression telle que par exemple un appel de fonction. La substitution est de ce fait beaucoup plus puissante qu'un transducteur fini : elle a la puissance de la machine de Turing. En contre-partie, elle est moins susceptible de précompilation et ne factorise pas les calculs.

Il existe des systèmes qui implémentent de façon exemplaire les machines finies et leurs opérations mais sont très sommaires du point de vue des structures de programmation. C'est le cas de Xfst (Xerox Finite State Tool). Xfst est utilisé surtout pour la morphologie et la phonologie (Beesley & Karttunen, 2003). Il dispose des expressions régulières habituelles étendues avec les règles de réécriture contextuelles. Des descriptions morphologiques ou phonologiques très élégantes ont été données sous forme de cascades de règles contextuelles, compilées sous forme d'une cascade de transducteurs. Mais le langage proposé est très pauvre : il n'y a pas réellement de variables, pas de modularité, pas de fonction, pas de boucle ni de conditionnelle. Pratiquement, on ne dispose que de la séquence qui permet d'exécuter des opérations l'une après l'autre.

D'autres système font le choix de se présenter comme KNG sous la forme d'une bibliothèque, mais dans la plupart des cas, ce sont des bibliothèque C++. Citons notamment `OpenFst` et Stuttgart Finite State Transducer tools (SFST) (Schmid, 2005). Les structures de programmation sont riches, mais au prix d'une complexité du langage hôte qui rend le développement plus difficile.

PyOpenFst [1] est comme KNG une couche encapsulant OpenFst dans un module Python. Comme OpenFst, il n'implémente pas les expressions régulières ni a fortiori les règles de réécriture contextuelle. Il n'offre pas non plus la même variété d'entrées-sorties.

KNG est un outil opérationnel mais encore expérimental. Au moment où cet article est écrit, il est utilisé pour réécrire partiellement SxPipe (Sagot & Boullier, 2005), une chaîne de traitement morpho-lexical. Dans cette chaîne, des composants KNG sont utilisés dans une cascade comportant également des scripts Perl et des programmes en C.

Nous prévoyons de diffuser prochainement KNG sous licence LGPL.

Références

BEESLEY K. R. & KARTTUNEN L. (2003). *Finite State Morphology*. CSLI Publications.

DEMAILLE A., DURET-LUTZ A., LESAINT F., LOMBARDY S., SAKAROVITCH J. & TERRONES F. (2009). An xml format proposal for the description of weighted automata, transducers and regular expressions. In *Proceedings of the 2009 Conference on Finite-State Methods and Natural Language Processing : Post-proceedings of the 7th International Workshop FSMNLP 2008*, p. 199–206, Amsterdam, The Netherlands.

KAPLAN R. M. & KAY M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, **20 :3**, 331–378.

MOHRI M., PEREIRA F. C. N. & RILEY M. (2002). Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, **16**(1), 69–88.

SAGOT B. & BOULLIER P. (2005). From Raw Corpus to Word Lattices : Robust Pre-parsing Processing with SxPipe. *Archives of Control Sciences*, **15**(4), 653–662.

SCHMID H. (2005). A programming language for finite state transducers. In A. YLI-JYRÄ, L. KARTTUNEN & J. KARHUMÄKI, Eds., *Finite-State Methods and Natural Language Processing*, volume 4002 of *Lecture Notes in Computer Science*, p. 308–309, Helsinki, Finland.

W3C (2012). State chart xml (scxml) : State machine notation for control abstraction w3c working draft 16 february 2012.

1. http ://code.google.com/p/pyopenfst/

21ème Traitement Automatique des Langues Naturelles, Marseille, 2014

Comparaison de deux outils d'analyse de corpus japonais pour l'aide au linguiste, Sagace et MeCab

Raoul Blin
CNRS-CRLAO, 131 Bd St.Michel, 75006 Paris
blin@ehess.fr

Résumé. L'objectif est de comparer deux outils d'analyse de corpus de textes bruts pour l'aide à la recherche en linguistique japonaise. Nous mesurons leur précision dans la tâche de comptage de chaînes de morphes. Les deux outils représentent chacun une approche spécifique. Le premier, un dispositif basé sur l'analyseur morphologique statistique MeCab, segmente et étiquette préalablement les phrases complètes. Le second compte les occurrences de la chaîne dans le texte en l'état. Les performances de Sagace sont globalement un peu inférieures mais la différence est moins importante qu'attendu. Du fait de leur facilité de mise en œuvre, les outils comme Sagace sans analyse morphologique préalable sont donc des outils malgré tout intéressants pour le linguiste.

Abstract. Our purpose is to compare two tools used to help linguists analyze large corpora of raw Japanese text. We measure their precision while counting strings of morphs. Each tool is representative of a specific approach. The first tool is based on the statistical morphological analyzer MeCab. It first tokenizes and POS tags the whole sentence before searching and counting strings. The second tool, Sagace, searches and counts within the text as it is. In accordance with our assumptions, Sagace performed slightly worse overall but the difference is not as marked as expected. Taking into account the needs of linguists, Sagace is nevertheless useful for many tasks.

Mots-clés : Japonais, Corpus, Analyseurs morphologique, MeCab, Sagace
Keywords: Japanese, Corpus, Morphological analyzer, MeCab, Sagace

Le linguiste travaillant sur le japonais écrit contemporain dispose de nombreux outils d'analyse de corpus, que ce soit pour dénombrer des collocations ou constituer des concordances. On peut séparer ces outils en deux groupes. Le premier rassemble les outils qui s'en tiennent à chercher un patron dans du texte brut. Les outils du second groupe procèdent en deux temps : le texte est d'abord segmenté en morphes et étiqueté en parties du discours, puis a lieu la recherche et le comptage de patrons. En japonais, à cause de l'absence de marques de frontière des mots et des nombreuses ambiguïtés graphiques, la seconde approche paraît la plus fiable. Il reste cependant à mesurer l'écart réel de qualité entre les dispositifs. Si cet écart est « raisonnable », alors la légèreté et la facilité de mise en oeuvre des premiers dispositifs pourraient leur donner un avantage, face à des dispositifs à base de segmentation préalable dont la mise en oeuvre est longue. En effet, les analyseurs morphologiques sont aujourd'hui tous statistiques et nécessitent un entraînement. Dans la pratique et en particulier en TAL, de nombreuses études utilisent les lexiques existants. Mais de nombreuses recherches en linguistique nécessitent de modifier ces lexiques, que ce soit les entrées lexicales elle-même ou bien les parties du discours Pour conserver une analyse morphologique statistique optimale, il faudrait à chaque modification du lexique réentraîner l'analyseur, ce qui demande un temps considérable incompatible avec les pratiques en linguistique.

Dans cet article, nous comparons les performances d'extraction et de comptage des noms communs, avec et sans analyse morphologique (statistique) préalable. Pour cela, nous utilisons des outils déjà disponibles, Sagace (Blin, 2012) et un dispositif basé sur l'analyseur morphologique statistique MeCab (Kudo, 2006). Dans la première partie, nous présentons les caractéristiques du japonais écrit qui sont susceptibles d'affecter les résultats des comptages automatiques effectués par ces dispositifs. La deuxième partie est consacrée à la description des outils et des ressources utilisés pour le comparatif. Les résultats de la comparaison figurent dans la troisième partie.

1 Rappel sur le japonais écrit

Nous rappelons quelques propriétés du japonais écrit qui sont susceptibles d'influencer l'analyse automatique des noms communs. Ces propriétés ont été largement décrites ailleurs de façon plus globale (voir par exemple en français (Yazawa, 2006)).

Le japonais utilise trois écritures : les hiragana et katakana que l'on peut qualifier de phonétiques (50 caractères de base pour chaque série) et un ensemble d'environ 2000 sinogrammes. Les noms communs empruntés au XXè S. aux langues étrangères sont écrits en katakana. Les autres peuvent indifféremment être transcrits avec l'une des trois écritures, voire avec un mélange. Il existe des « recommandations » officielles d'emplois des écritures. Elles sont plus ou moins respectées selon les milieux sociolinguistiques. La presse s'y conforme assez strictement. Nous estimons qu'il en va de même pour les brevets. Par contre, les blogs n'ont pas de contraintes mais nous estimons que dans leur immense majorité, les pratiques sont proches de la norme officielle (une raison est que les textes sont saisis au clavier à l'aide de logiciel de transcription qui par défaut appliquent les recommandations officielles).

Dans les lexiques associés aux analyseurs morphologiques automatiques et utilisés ici, les graphies les plus courantes d'un mot font en général l'objet d'une entrée propre. Par exemple le mot *rôka* (« couloir ») est enregistré trois fois dans le lexique UniDic (voir plus bas), avec trois graphies différentes : sinogrammes 廊下, hiragana ろうか, mélange des deux ろう下. D'autres variantes sont possibles, notamment en utilisant les katakana. Ces variantes n'étant pas répertoriées, elles ne seront pas détectées dans les textes par les analyseurs que nous utilisons. Il s'ensuit que le nombre de mots reconnus peut être minimisé. Cependant, les variantes non lexicalisées sont celles qui apparaissent rarement, sauf dans des textes très spécifiques. Leur omission a donc très peu d'impact dans les comptages de textes « ordinaires » comme ceux utilisés pour le présent travail.

Comme on le voit dans l'exemple précédent, le nombre de caractères qui composent un mot peut changer en fonction du choix de l'écriture. C'est ce qui explique qu'à nombre égal de mots, la longueur de deux textes mesurée en nombre de caractères peut varier si ils utilisent des écritures différentes. Sauf pour quelques noms, l'écriture en sinogrammes est systématiquement une des plus courtes.

Les noms communs japonais sont invariables morphologiquement. Seule leur graphie varie. En conséquence, pour cette catégorie, une entrée du lexique correspond à la fois à un mot et à un morphe. Dans les comptages, il serait possible de regrouper les variantes graphiques d'un mot, de sorte à les considérer comme les occurrences d'un seul mot. Par exemple les trois graphies de *rôka* seraient considérées comme des occurrence du mot « couloir ». Nous ne l'avons pas fait car nous ne savons pas quel type d'erreur peut survenir dans la reconnaissance du mot ni l'impact que cela peut avoir sur les comptages, compte tenu des différences entre les lexiques. Nous mesurons donc les occurrences des morphes/entrées lexicales indépendamment les uns des autres. Nous utiliserons désormais le terme de morphe.

Il existe de nombreux homographes. En particulier, phénomène peu souligné, de très nombreux noms communs peuvent être utilisés comme patronymes. Aucune marque graphique ne distingue les deux emplois. Par exemple 森 (*mori*, « forêt ») est un patronyme très répandu. La phrase *mori*$_{N}$ *wo*$_{O}$ *miru*$_{V}$ peut tout aussi bien signifier « Regarder$_{V}$ Mori$_{N}$. » que « Regarder$_{V}$ la forêt$_{N}$. ». Des ambiguïtés surviennent aussi avec les homophones lorsqu'ils sont transcrits phonétiquement, comme par exemple はし (*hashi*), « pont » ou « baguette(s) » (couvert). Avec les trois lexiques de la présente étude, MeCab propose à tort comme première interprétation « baguette » pour l'occurrence de *hasi* dans la phrase *kuruma*$_{N}$ *de*$_{en}$, *hashi wo*$_{O}$ *watatta*$_{V}$ (« En voiture$_{N}$, [j']ai traversé [le] *hashi* »). Des techniques existent pour lever l'ambigüité (Tokunaga, 2012) mais ne sont pas intégrées dans les analyseurs morphologiques. Des erreurs de comptage peuvent s'ensuivre de la confusion des morphes, aussi bien dans le sens d'un plus grand bruit que d'un plus grand silence. L'ajout d'une couche d'analyse, notamment des relations de dépendances, serait possible et utile pour désambiguïser, mais alourdirait considérablement le dispositif tout en risquant d'introduire d'autres erreurs.

A l'exclusion des textes pour enfants, le japonais ne sépare pas graphiquement les « mots » ni les morphes. A cause de leur faible nombre d'occurrences, les signes de ponctuation n'aident que très peu. Quant à l'alternance des écritures, elle est trop aléatoire pour être utilisée massivement. En l'absence de séparation, un morphe peut être détecté à tort à cheval sur deux morphes. Ainsi, dans l'exemple 1 ci-dessous, [*butugo*] (« parole de Buddha ») est détecté à tort à cheval sur les morphes <*niti hutu*> (« franco-japonais ») et <*gogaku gakkai*> (« Société de langues »). Un contrôle sur le contexte des noms permet aisément d'éviter ce type d'erreurs. Une analyse morphologique statistique effectue un tel contrôle. Sagace limitera les erreurs grâce à la stratégie de priorité donnée au morphe le plus long : par préférence du morphe le

plus long, on privilégiera le mot de deux caractères *nitihutu* comme mot de tête, plutôt que celui à un caractère *niti* (« jour »). Cela exclut alors le découpage *niti+hutugo*. La longueur du morphe étant mesurée en nombre de caractères, on comprend que son succès dera tributaire des conventions d'écritures adoptées dans le texte. Un morphe peut aussi être détecté à l'intérieur d'un autre morphe comme dans l'exemple 2 : le morphe *nihonjin* (« Japonais ») « contient » le mot *hi* (« jour »). Dans cet exemple, deux analyses sont possibles et prendre en compte le contexte immédiat ne permet pas de détecter la meilleure analyse. Il faut recourir à d'autres techniques, basées sur les associations sémantiques de « mots », mais qui ne sont pas intégrées aux analyseurs morphologiques. On voit ici que la stratégie de préférence du mot le plus long privilégiera systématiquement l'interprétation « *nihonjin* ». Sagace ne détectera donc pas l'interprétation 1.

Ex 1 : <日 [仏 > < 語] 学 > <学 会>
<*niti* [*hutu*> < *go*] *gaku*> <*gak kai*>
Société franco-japonaise de langues

Ex 2 : 1) その/日 // 本人 が 来た。
sono / hi // hon'nin ga kita
« Ce jour là, la personne elle même est venue. »

2) その/日本人// が 来た 。
sono / nihonjin // ga kita
« Ce Japonais est venu. »

2 Dispositifs d'analyse et ressources utilisés pour la comparaison

Pour comparer les deux approches, avec ou sans analyse morphologique (statistique) préalable du texte, nous avons choisi Sagace et un dispositif basé sur l'analyseur morphologique MeCab. Les deux ont en commun de prendre du texte brut en entrée. Ils sont en ligne de commande, et sont open source et libres. Des chercheurs peuvent ainsi aisément les adapter pour des nouveaux besoins et intégrés à d'autres dispositifs. Dans cette section, nous présentons leurs principales caractéristiques. Comme les performances des dispositifs sont très dépendantes du contenu des lexiques associés, nous consacrerons une sous-section aux lexiques utilisés pour les tests. Aucun des analyseurs utilisés ne comptabilise les variantes graphiques de morphes non répertoriées dans le lexique. Le silence s'en trouve donc augmenté.

2.1 Sagace

Parmi les outils qui ne procèdent pas à une analyse morphologique préalable, ceux qui utilisent les caractères comme unités (`grep` etc.) sont d'un intérêt très limité pour le linguiste. Il existe d'autres outils, qui prennent comme unité de travail les morphes listés dans des lexiques. Nous avons choisi comme représentant de cette approche Sagace car il est disponible avec un lexique au format adéquat et qu'il est utilisé de longue date pour le japonais. D'autres outils peuvent faire des recherches comparables (par exemple Unitex) mais ils auraient nécessité de construire un lexique. Dans la limite de nos investigations, nous n'avons pas trouvé d'outil commercial équivalent.

Sagace est conçu dès l'origine comme un outil d'aide à la recherche en linguistique. C'est un outil générique exploitable pour n'importe quelle langue. Il peut produire des concordances ou lister et compter des collocations. Il se veut plus adapté au linguiste en permettant une manipulation très rapide et souple des lexiques grâce à un langage (de type propositionnel) de description des catégories avec lequel il est possible de créer « à la volée » des nouvelles catégories, sur la base des catégories existantes dans le lexique, et sans intervenir dans celui-ci. Autre aspect pensé pour le linguiste : il est tout en un. Il contient une interface d'interrogation, un moteur de recherches et de comptage, et un compilateur de lexique. Il ne nécessite pas de logiciels externes (contrairement aux dispositifs avec analyse morphologique préalable), ni de segmentation préalable (comme c'est le cas de Himawari (Yamaguchi, 2011)). Mais surtout, il ne nécessite pas non plus d'entraînement comme les logiciels statistiques.

Lors de l'analyse, Sagace découpe le texte en « segments », délimités par un morphe *ad hoc* choisi par l'utilisateur. Pour la présente comparaison, nous prenons comme segment la phrase. Les segments dépassant une certaine longueur sont rejetés pour éviter des problèmes de mémoire. Les occurrences de patrons recherchés qui sont présents dans les segments rejetés ne sont donc pas pris en compte par Sagace alors qu'ils seront traités par les autres dispositifs. De ce fait, le silence dans les résultats de Sagace pourrait être supérieur. Lorsque sur une position Sagace détecte plusieurs morphes possibles, il privilégie le morphe le plus long. Cette méthode usuelle en analyse morphologique du japonais est souvent gagnante, mais peut provoquer des erreurs.

2.2 Dispositif basé sur MeCab

Les dispositifs qui procèdent à une analyse morphologique préalable (segmentation et étiquetage des parties du discours) sont des assemblages composés d'un analyseur morphologique et d'un outil de comptage qui fait aussi interface pour l'interrogation. A notre connaissance, il n'existe pas de dispositif « tout en un ». Pour des questions pratiques, nous ne pouvions pas ici évaluer les performances de toutes les combinaisons possibles de tous les analyseurs morphologiques et outils de comptage. Pour des raisons de disponibilité et de réputation, nous avons choisi l'analyseur morphologique statistique MeCab, parmi les plus utilisés du moment JUMAN (Kurohashi et al., 1994), KyTea (Neubig et al., 2011). Il est disponible sous forme de paquets pour les systèmes de type Unix. Le logiciel est en plus distribué avec plusieurs lexiques japonais libres eux aussi (voir section suivante). Pour simplifier la présentation, désormais, nous désignerons la combinaison de MeCab, d'un lexique et d'un outil de comptage par le terme « dispositif MeCab ». MeCab utilise un modèle de Markov d'ordre 1. Les paramètres du modèle, inscrits dans le lexique, sont estimés par apprentissage. MeCab prend comme segment de travail toute chaîne de caractères finissant par un saut à la ligne. Dans les textes bruts il arrive que des sauts à la ligne apparaissent à l'intérieur d'une phrase. Les analyses peuvent être perturbées. Nous estimons que l'erreur induite augmente le silence mais pas au point de modifier sensiblement les résultats.

2.3 Les lexiques

Les performances des dispositifs d'analyse varient en fonction des lexiques. C'est la raison pour laquelle nous avons mené les tests avec les lexiques disponibles pour MeCab. UniDic[1] contient 756 463 entrées lexicales, noms propres compris, ce qui en fait le lexique le plus volumineux parmi les trois utilisés ici. Contrairement aux deux autres lexiques, le choix des entrées lexicales/morphes a été argumenté et explicité (The UniDic Consortium, 2012). Le principe est de lexicaliser les plus petites unités morphosémantiques possibles. A coté de cela, les concepteurs et mainteneurs de MeCab recommandent[2] un second lexique, IPAdic[3] (392 126 entrées). Enfin, il faut compter avec un troisième lexique, Jumandic[4] (515 996 entrées). Dans ces trois lexiques figurent diverses informations : lecture, écriture usuelle du morphe si il s'agit d'une variante graphique, et partie du discours. Figurent aussi les paramètres du modèle statistique utilisé. Les paramètres ont été estimés par apprentissage sur des corpus différents, respectivement le BCCWJ data core (voir section suivante), le corpus IPA (Hashimoto, 1995) et le corpus de Kyodai (Kawahara et al., 2002). Avec Sagace, nous avons utilisé une version modifiée du naist-jdic[5] (485 000 entrées), qui est très proche du lexique IPAdic. Nous estimons que les lexiques possèdent un nombre de noms communs à peu près identique, soit environ 90 000 si l'on se base sur le naist-jdic.

2.4 Corpus

Les tests ont été effectués sur le Balanced Corpus of Written Japanese - Data Core, version 2009 (Maekawa, 2009). Il s'agit d'un corpus segmenté manuellement en morphes et étiqueté (au format XML) en reprenant les morphes et parties du discours du lexique UniDic 1.3.12 . Nous en avons tiré un texte brut sans les balises. Ce corpus s'est imposé ici car il est devenu aujourd'hui une référence en linguistique. Il est aussi utilisé en traitement automatique (voir des références dans (Mori et al., 2014)). D'autres corpus, dont celui de Kyodai ou YACIS (Ptaszynski et al., 2012), sont disponibles mais leur usage reste limité au traitement automatique. Le corpus comprend des extraits de blogs, de journaux économiques, de brevets et des phrases exemples de dictionnaires. Nous estimons que les conventions d'écriture standard sont globalement respectées. En particulier, la majorité des noms communs sont transcrits en sinogrammes lorsqu'ils peuvent l'être. Le recours à la transcription phonétique et les risques d'ambiguïtés dus à l'homophonie en sont diminués. Le corpus brut ne contient pas de phrases d'une longueur excédant les limites imposées par Sagace. Aucune phrase n'est donc rejetée par ce dernier. Certaines phrases contiennent des sauts de lignes. Cela peut affecter l'analyse de MeCab qui n'a pas été entrainé sur des corpus contenant de tels sauts de lignes.

---

1 unidic-mecab 2.1.2

2 mecab.googlecode.com/svn/trunk/mecab/doc/index.html

3 mecab-ipadic.x86_64 ; 2.7.0.20070801-6.fc18.1 .

4 mecab-jumandic.x86_64, ref ; 5.1.20070304-7.fc18 .

5 mecab-naist-jdic-0.4.3-20080812

3 Taux d'accord

Nous comparons les taux d'accord entre les analyses automatiques et l'analyse manuelle dans une tâche d'extraction d'un morphe isolé (nom commun) et dans une tâche d'extraction d'une chaîne de trois morphes (un nom commun et son contexte gauche et droite). L'extraction et le comptage sont effectués avec les lexiques en l'état, donc avec les 90 000 noms communs. Mais la comparaison des résultats n'est effectuée que sur les noms communs présents à la fois dans les trois lexiques et dans le texte segmenté manuellement. Les erreurs sur les autres noms ne sont donc pas prises en compte, ce qui pourrait minimiser artificiellement le bruit. Il en va de même avec les signes de ponctuation et les particules qui sont utilisées pour définir le contexte du nom dans le deuxième test.

Nous comptons tout d'abord les occurrences de 4 000 noms communs, sans poser de contraintes sur le contexte. Les résultats (table 1) sont conformes aux attentes. Mecab est meilleur, avec bien sûr un avantage pour l'analyse avec le lexique Unidic. Les écarts types montrent que les variations sont plus importantes avec Sagace. Nous faisons l'hypothèse que Sagace fait sensiblement plus d'erreurs sur les morphes courts et rédigés en caractères phonétiques que sur les morphes longs, en sinogrammes. La différence entre Sagace et les dispositifs MeCab, si elle n'est pas négligeable, n'est cependant pas considérable, en particulier par rapport aux deux dispositifs MeCab qui n'ont pas été entraînés sur le corpus étudié et dont les résultats sont de ce fait plus représentatifs des performances réelles des dispositifs.

	RAPPEL		PRECISION		F-MESURE
	moyenne	écart type	moyenne	écart type	
Sagace	89.34	.24	83.86	.39	.865
MeCab+ IPAdic	96.85	0.11	96.03	0.13	.964
MeCab + Jumandic	95.52	0.14	97.37	0.11	.964
MeCab + UniDic	99.09	0.06	98.78	0.08	.989

TABLE 1 : Taux d'accord des analyses de Sagace et MeCab par rapport à une analyse manuelle basée sur le lexique UniDic ; comptage des noms communs, sans contrainte sur le contexte immédiat.

Nous comparons cette fois-ci les taux d'accord dans la tâche d'extraction des noms communs mais en posant des contraintes sur leur contexte immédiat. Nous imposons la présence de particules ou signes de ponctuation (44 morphes et symboles en tout) de chaque coté du nom commun.

	RAPPEL		PRECISION		F-MESURE	
	moyenne	écart type	moyenne	écart type		Diff. avec Table 1
Sagace	85.89	.28	89,28	.31	.875	+ 0.010
MeCab+ IPAdic	83.98	.31	96.06	.17	.896	- 0.068
MeCab + Jumandic	95.07	.20	89.60	.26	.922	- 0.041
MeCab + UniDic	88.42	.26	99.19	.08	.934	- 0.054

TABLE 2 : Taux d'accord des analyses de Sagace et MeCab par rapport à une analyse manuelle basée sur le lexique UniDic ; comptage des noms communs, avec contraintes sur le contexte immédiat.

Au regard des f-mesures (Table 2) l'écart de performances entre Sagace et les dispositifs MeCab s'est réduit, mais du fait de mouvements inverses : Sagace est meilleur que pour une recherche de morphe isolé tandis que les dispositifs MeCab sont moins performants. Pour Sagace, l'amélioration est due à l'allongement du patron cherché, et à la réduction des risques d'ambiguïtés qui s'ensuit. Sagace rejoint même MeCab sur certains résultats. Pour MeCab, la

baisse est plus sensible avec le lexique IPAdic pourtant conseillé par les auteurs du logiciel. Une explication est peut-être dans le choix des parties du discours ou du corpus d'entraînement. Une étude plus approfondie reste à faire.

4 Conclusion

Dans cette étude, nous avons comparé les performances de deux dispositifs d'analyse de corpus à destination des linguistes du japonais, pour l'exécution de tâches tout à fait usuelle en linguistique : le dénombrage de mots ou de morphes. Ces dispositifs ont été choisis comme représentatifs de deux types de traitement des corpus : une approche « légère » qui s'en tient à la recherche de patrons sans analyse générale des phrases, et une approche complète, plus lourde avec analyse morphologique du texte analysé. Les résultats montrent que, à condition d'éviter certains types de recherches, les performances du premier dispositif sont raisonnables. Si l'on tient en plus compte de l'ergonomie et en particulier la facilité de mise en oeuvre et la vitesse de traitement, il s'agit d'un outil tout à fait suffisant pour le débroussaillage de corpus et l'aide à l'analyse linguistique.

Références

BLIN R. (2012). SAGACE v4.2.0.

HASHIMOTO M. (1995). Building a Corpus at IPA. 全国大会講演論文集 *51*, 35–36.

KAWAHARA D., KUROHASHI S., and HASIDA K. (2002). Construction of a Japanese Relevance-tagged Corpus. In Proceedings of the 3rd International Conference on Language Resources and Evaluation, pp. 2008–2013.

KUDO T. (2006). MeCab: yet another part-of-speech and morphological analyzer.

KUROHASHI S., NAKAMURA T., MATSUMOTO Y., and NAGAO M. (1994). Improvements of Japanese Morphological Analyzer JUMAN. In Proceedings of The International Workshop on Sharable Natural Language Resources, pp. 22–28.

MAEKAWA K. (2009). Daiyousei wo yû suru daikibo nihongo kakikotoba kôpasu (<tokushyû> nihongo kôpasu) [Compilation d'un corpus équilibré de textes contemporains en japonais. J. Jpn. Soc. Artif. Intell. *24*, 612–622.

MORI S., OGURA H., and SASADA T. (2014). A Japanese Word Dependency Corpus. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), (Reykjavik, Iceland: European Language Resources Association (ELRA)), pp. 753–758.

NEUBIG G., NAKATA Y., and MORI S. (2011). Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, (Stroudsburg, PA, USA: Association for Computational Linguistics), pp. 529–533.

PTASZYNSKI M., RZEPKA R., ARAKI K., and MOMOUCHI Y. (2012). Automatically Annotating a Five-billion-word Corpus of Japanese Blogs for Affect and Sentiment Analysis. In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, (Stroudsburg, PA, USA: Association for Computational Linguistics), pp. 89–98.

THE UNIDIC CONSORTIUM (2012). unidic-mecab ver. 2.1.1 Users Manual.

TOKUNAGA T. (2012). Nihongo nyuuryoku wo sasaeru gijutu [Technologies behind Japanese input systems].

YAMAGUCHI M. (2011). zenbun kensaku sisutemu "Himawari" riyousha manyuaru vers.1.3 [Système d'extraction "Himawari"; Manuel de l'utilisateur vers.1.3].

YAZAWA M. (2006). Traitement de texte en japonais et enseignement des kanji. In Langue, Lecture et École Au Japon, (Arles: Christian Galan et Jacques Fijalkow), pp. 97–137.

Un concordancier multi-niveaux et multimédia pour des corpus oraux

Giulia Barreca[1] George Christodoulides[2]
(1) Laboratoire MoDyCo, CNRS, Université Paris Ouest Nanterre La Défense
200, avenue de la République, FR-92001 Nanterre, France
(2) Centre Valibel, Institut Langue & Communication, Université de Louvain,
Place Blaise Pascal 1, 1348 Louvain-la-Neuve, Belgique
giulia.barreca@gmail.com, george@mycontent.gr

Résumé. Les concordanciers jouent depuis longtemps un rôle important dans l'analyse des corpus linguistiques, tout comme dans les domaines de la philologie, de la littérature, de la traduction et de l'enseignement des langues. Toutefois, il existe peu de concordanciers qui soient capables d'associer des annotations à plusieurs niveaux et synchronisées avec le signal sonore. L'essor des grands corpus de français parlé introduit une augmentation des exigences au niveau de la performance. Dans ce travail à caractère préliminaire, nous avons développé un prototype de concordancier multi-niveaux et multimédia, que nous avons testé sur le corpus de français parlé du projet Phonologie du Français Contemporain (PFC, 1,5 million de tokens de transcription alignée au niveau de l'énoncé). L'outil permet non seulement d'enrichir les résultats des concordances grâce aux données relevant de plusieurs couches d'annotation du corpus (annotation morphosyntaxique, lemme, codage de la *liaison*, codage du *schwa* etc.), mais aussi d'élargir les modalités d'accès au corpus.

Abstract. Concordances have always played an important role in the analysis of language corpora, for studies in humanities, literature, linguistics, translation and language teaching. However, very few of the available systems support multi-level queries against a richly-annotated, sound-aligned spoken corpus. The rapid growth in the development of spoken corpora, particularly for French, increases the need for scalable, high-performance solutions. We present the preliminary results of our project to develop a multi-level multimedia concordancer for spoken language corpora. We test our prototype on the PFC corpus of spoken French (1.5 million tokens, transcriptions aligned to the utterance level). Our tool allows researchers to query the corpus and produce concordances correlating several annotation levels (part-of-speech tags, lemmas, annotation of phonological phenomena such as the *liaison* and *schwa*, etc.) while allowing for multi-modal access to the associated sound recordings and other data.

Mots-clés : concordancier, annotation multi-niveaux, linguistique de corpus, didactique du FLE
Keywords: concordance tool, multi-level annotation, corpus linguistics, French language teaching

1 Introduction

L'utilisation d'outils informatiques tels que les concordanciers, qui permettent d'extraire des collocations de mots et leur contexte, est fréquente dans la communauté des chercheurs en sciences du langage. La taille de corpus actuellement disponibles est en augmentation constate, de même que les schèmes d'annotation, ce qui entraîne la création de nouveaux systèmes pour interroger ces données. Dans les pages qui suivent, après avoir décrit brièvement quelques-uns des travaux menés sur les concordanciers, nous présentons un prototype de concordancier multi-niveaux et multimédia que nous sommes en train de développer et tester sur les données du corpus de français parlé PFC (Phonologie du Français Contemporain, cf. Durand et al. 2002). Notre objectif est de nous rapprocher le plus possible des besoins des linguistes et des apprenants, qui souhaitent dépouiller de façon rapide et précise les contextes d'occurrences d'un lexème ou d'une séquence de lexèmes. Notre outil, contrairement à d'autres concordanciers, intègre en effet l'écoute des extraits d'enregistrements au travail de recherche des concordances. Il permet de lancer des requêtes croisant plusieurs couches (niveaux) d'annotation alignées temporellement avec le signal. Le moteur de recherche du concordancier construit de manière dynamique des requêtes SQL vers une base de données relationnelle, dans laquelle sont stockées les annotations. Dans le cas du corpus PFC, ces annotations associées à la transcription sont de nature morphosyntaxique (POS, lemme) et phonologique (codage des phénomènes de la liaison et du schwa). Nous montrons les applications possibles de l'outil dans les différents domaines de la linguistique et d'enseignement du FLE, son architecture et quelques-unes de ses fonctionnalités.

2 Synthèse des études précédentes

Le concordancier a toujours été un outil de grand intérêt pour l'analyse des contextes d'emploi des lexèmes dans des corpus oraux ou écrits. Ses applications théoriques et pratiques sont nombreuses, et intéressent plusieurs disciplines : philologie, littérature (Caballero 1999, Magri-Mourgues 2006), syntaxe et sémantique (Gross 2000), traduction (Jacquet-Pfau 1994) et didactique des langues (Tognini-Bonelli 2001). Pourtant, comme le soulignent Pincemin et al. (2006), la plupart des concordanciers disponibles n'exploitent pas les informations linguistiques introduites par l'annotation des corpus (p. ex. catégories morphosyntaxiques, lemmatisation, etc.). Les résultats produits par ce genre de concordancier n'aboutissent qu'à de simples collections d'occurrences (p.ex. Lextutor et Lexiqum). Plusieurs concordanciers existent déjà pour les corpus écrits : parmi ceux-ci nous pouvons citer Condor (développé par le Laboratoire lorrain de recherche en informatique – LORIA), qui permet d'effectuer des recherches simples ou de cooccurrences utilisant le mot-forme ou le lemme, et d'indexer les formes présentes dans le corpus. La version publiquement disponible propose une collection des corpus écrits, pour la plupart littéraires. Le concordancier TXM (Heiden et al. 2010) permet de produire des concordances KWIC (*keyword in context*) à partir des propriétés des mots (ex. lemme, catégorie POS) ou de lancer de requêtes plus complexes grâce à une syntaxe propre à son moteur de recherche.

L'intervention des informations de nature linguistique, et donc l'affinement des résultats des concordances, demande un traitement préalable du corpus qui est souvent très couteux, comme c'est par exemple le cas du corpus French Treebank (Abeillé 2003). Pour cette raison, de nombreux concordanciers lemmatisés font appel à des dictionnaires. Le recours à ce type de ressources linguistiques permet notamment de mettre en relation les occurrences du texte avec les lemmes et les formes fléchies contenues dans les dictionnaires. Parmi les concordanciers qui utilisent ce type d'outils linguistiques, nous pouvons citer *Unitex* (Paumier 2002) et *Stella* (Bernard et al. 2002).

Toutefois, l'application des concordanciers conçus pour l'écrit à l'analyse des corpus oraux pose un problème de fond : ces outils ne permettent pas de tenir compte de l'alignement des annotations linguistiques avec le signal sonore. Les requêtes possibles sont ainsi limitées à ce qu'on peut représenter avec les conventions de transcription choisies. À notre connaissance, les concordanciers lemmatisés pour l'analyse de corpus de français parlé sont peu nombreux. Mis à part le concordancier de la base lexicale *Lexique 3* (New 2006), basé sur les transcriptions des sous-titrages de films, les autres concordanciers disponibles pour les corpus oraux[1] ne permettent d'obtenir que des relevés des occurrences et des données statistiques sur leurs collocations. Dans ce contexte, nous considérons qu'un concordancier multimédia spécifique pour l'analyse d'un corpus oral pourrait permettre aux utilisateurs de mieux prendre en compte l'articulation des différents niveaux d'annotation dans le temps et d'exploiter les résultats des requêtes tout en gardant le lien avec le signal sonore.

3 Objectif

Notre objectif est de développer un concordancier multi-niveaux et multimédia, exploitable pour des recherches linguistiques en français parlé. Pour ce faire nous avons utilisé le corpus PFC (Phonologie du Français Contemporain) (Durand et al. 2002) annoté en morphosyntaxe (POS et lemmes) à l'aide de l'étiqueteur *DisMo* (Christodoulides et al. 2014). Ce corpus, qui se compose de 36 points d'enquête pour un total de 394 locuteurs, constitue une ressource de grand intérêt pour tout linguiste intéressé au français parlé, grâce à la variété des origines géographiques des locuteurs enregistrés, des styles et de registres de parole, mais également des annotations des phénomènes phonologiques spécifiques au français parlé (liaison, schwa).

Nous avons essayé de créer un outil qui permette établir une articulation entre ces différents niveaux d'analyse, et qui puisse ainsi offrir une observation linguistique plus précise des contextes d'occurrence d'un phénomène linguistique donné (phonologique, syntaxique, lexicale ou prosodique). L'ensemble de ces contextes regroupés peut permettre de révéler la présence de régularités liées à certaines combinaisons sémantiques, syntaxiques et discursives des mots. Plusieurs études ont déjà mis en évidence l'influence de la fréquence sur les représentations cognitives (Bybee & Hopper 2001), sur la morphologie et la syntaxe (Bybee & Thompson 1997), sur la phonologie (Bybee 2001), sur l'acquisition (Tomasello 2000) et évidemment sur les procès de grammaticalisation (Bybee & Scheibman 1999). En particulier cette dernière application pourrait être d'un intérêt particulier dans l'étude du procès de pragmaticalisation (Dostie 2004) des marqueurs discursifs du français parlé.

[1] Parmi lesquels ceux des corpus CFPP2000 (Corpus de Français Parlé Parisien), CLAPI (Corpus de Langue Parlée en Interaction) et OFROM (Corpus Oral de Français parlé en Suisse Romande).

L'utilisation d'un concordancier lemmatisé sur un corpus annoté pourrait aussi permettre d'extraire de façon automatique les contextes d'occurrences des marqueurs discursifs, et de repérer leurs positions prototypiques ainsi que leurs combinaisons possibles et effectives (p. ex. *« enfin bon »*), ceci afin de mieux saisir leurs fonctions pragmatiques en discours. De plus, l'utilisation d'un concordancier lemmatisé permettrait de développer une analyse multidimensionnelle des phénomènes phonologiques spécifiques au français parlé, phénomènes dont la réalisation est influencée par des facteurs relevant de l'articulation de plusieurs dimensions linguistiques (phonologique, syntaxique, lexicale, prosodique etc.). C'est le cas par exemple du phénomène de la liaison. Plus précisément une telle analyse permettrait de vérifier l'influence des facteurs lexicaux (fréquence des mots liaisonnés et de leur cooccurrence) sur la réalisation de la liaison, tout en donnant accès aux données nécessaires pour une analyse phonologique et syntaxique de la séquence liaisonnée. L'utilisation d'un concordancier lemmatisé pourrait aussi permettre de mieux analyser le rôle joué par la fréquence d'usage des mots et leur fréquence distributionnelle dans certaines constructions où ni la liaison, ni le figement, ne sont prévisibles. C'est le cas par exemple des séquences *temps en temps* vs *temps/à autres* (Laks 2005). Nous considérons de façon plus générale qu'un tel outil permettra de placer la notion d'usage (Bybee & Hopper 2001) au centre de la réflexion linguistique.

En ce qui concerne la didactique du FLE, l'application d'un concondancier lemmatisé du français parlé constituerait une ressource précieuse pour l'enseignement, favorisant un apprentissage basé sur des données authentiques et conçu comme une recherche (*data-driven learning*), dans la lignée des travaux réalisés dans le cadre du sous-projet PFC-EF (PFC Enseignement du Français, Detey et al. 2010). Une telle approche fournirait aux apprenants une observation directe de la langue, qui aiderait à rapprocher les productions des apprenants des usages réels des natifs. Dans les paragraphes suivants, nous illustrons plus en détails le fonctionnement du concordancier.

4 Architecture du système

Le concordancier fonctionne à partir d'une base de données relationnelle (SQLite, MySQL ou PostgreSQL), dans laquelle les annotations du corpus à plusieurs niveaux sont représentées sous le schéma du logiciel *Praaline* (Christodoulides 2014). Chaque niveau d'analyse correspond à une table, et chaque annotation de ce niveau à une colonne de la table. Des identifiants uniques des échantillons du corpus (`AnnotationID` et des locuteurs (`SpeakerID`) renvoient aux métadonnées. Les intervalles temporels sont représentés avec deux nombres entiers de 64 bits, en nanosecondes (`tMin`, `tMax`). Ce simple schéma permet de représenter des relations hiérarchiques entre deux ou plusieurs couches d'annotation, ainsi que les éventuels chevauchements sur la même couche. Il permet aussi de profiter du système d'indexation de la base de données pour améliorer la performance des requêtes, ou appliquer des restrictions sur les annotations (p. ex. un vocabulaire restreint au jeu d'étiquettes morphosyntaxiques est appliqué au champ `pos_min`). Un extrait du schéma est affiché ci-dessous (Figure 1).

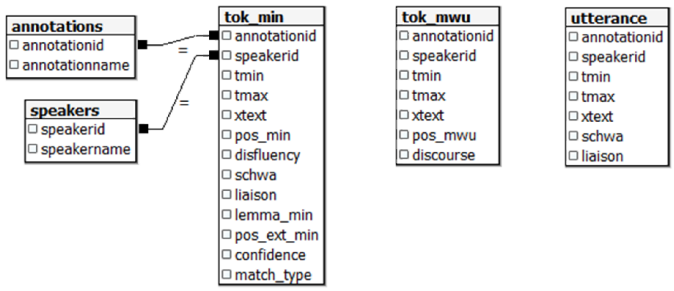


FIGURE 1 Extrait du schéma de la base de données. Les tables qui représentent les couches d'annotation (`tok_min`, `tok_mwu`, `utterance`) ont comme clé primaire la triplette (`annotationID`, `speakerID`, `tMin`).

Plusieurs méthodes ont été proposées pour le stockage des annotations d'un corpus, notamment sous la forme de fichiers XML, de bases de données NoSQL, ou même de structures de données propres à des logiciels d'annotation. Notre choix d'une base de données relationnelle est motivé par le fait que les annotations textuelles du corpus sont essentiellement structurées. Les indices temporels permettent au logiciel d'exploitation de corréler ces annotations avec le signal sonore (qui est stocké au format WAV), et avec d'autres annotations stockées au format HDF5. Ce dernier est le format de préférence pour les annotations qui impliquent un grand nombre de vecteurs à plusieurs dimensions, avec une fréquence d'échantillonnage élevée (p. ex. une série de mesures acoustiques et prosodiques).

Le logiciel que nous avons développé permet de construire de manière dynamique les requêtes SQL adéquates, selon les choix de l'utilisateur. Par exemple, la requête suivante renvoie les annotations associés à tous les 4-grammes « c'est-à-dire » (temps d'exécution : 649 ms sous SQLite 3.8.1, CPU i7 2.4 GHz, mémoire 24 Go) :

```
SELECT * FROM tok_min AS T1, tok_min AS T2, tok_min AS T3, tok_min AS T4
WHERE   T1.AnnotationID = T2.AnnotationID AND T1.SpeakerID = T2.SpeakerID AND T1.tMax = T2.tMin
 AND    T2.AnnotationID = T3.AnnotationID AND T2.SpeakerID = T3.SpeakerID AND T2.tMax = T3.tMin
 AND    T3.AnnotationID = T4.AnnotationID AND T3.SpeakerID = T4.SpeakerID AND T3.tMax = T4.tMin
 AND    T1.xText = 'c''' AND T2.xText = 'est' AND T3.xText = 'à' AND T4.xText = 'dire'
```

Une interface graphique permet à l'utilisateur de définir sa requête dans le corpus, qui est traduite à des requêtes de la base de données. Le système du moteur de recherche et du concordancier est indépendant du corpus : la structure de l'annotation est définie préalablement, et le schéma de la base de données est dynamique (modifié par le logiciel quand une couche d'annotation est ajoutée ou supprimée). L'outil utilise cette définition afin de créer les jonctions entre les tables qui représentent les différentes couches d'annotation.

5 Illustration: recherche de concordances lemmatisées et multimédia

Dans les paragraphes suivants nous allons présenter une démonstration des requêtes qui peuvent être réalisées à partir du prototype de notre concordancier. Nous allons montrer des extraits des tableaux des occurrences résultant de trois différents types de recherche: (a) par lexème; (b) par lemme; (c) par chaîne de caractères. Enfin nous illustrerons les résultats de la recherche de la cooccurrence de la séquence *je suis* et un exemple de l'affichage des résultats des concordances multimédia correspondantes.

Dans les exemples ci-dessous nous pouvons observer les résultats de la recherche de *suis* en tant que lexème et en tant que chaîne de caractères et en tant que lemme du verbe *être*. Les occurrences calculées sont présentées de façon synthétique par des tableaux avec, en colonne, de gauche à droite : le lexème, l'étiquette POS, le lemme, le nombre d'occurrences, la fréquence du lexème et la fréquence du lemme. De plus, chaque tableau est précédé par l'indication du nombre total des occurrences toute catégories confondues.

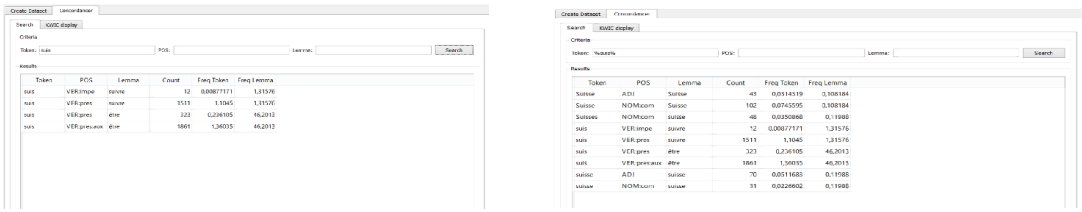


FIGURE 2 Accès à la concordance par tableau des résultats synthétiques de la requête du lexème *suis* (gauche) et de la chaîne de caractères %suis% (droite)

Il en va de même pour la recherche des cooccurrences par lexème et par lemme. Dans l'exemple ci-dessous, nous observons pour la séquence *je suis* deux différentes typologies de cooccurrences selon le lemme correspondant (*être* vs *suivre*).

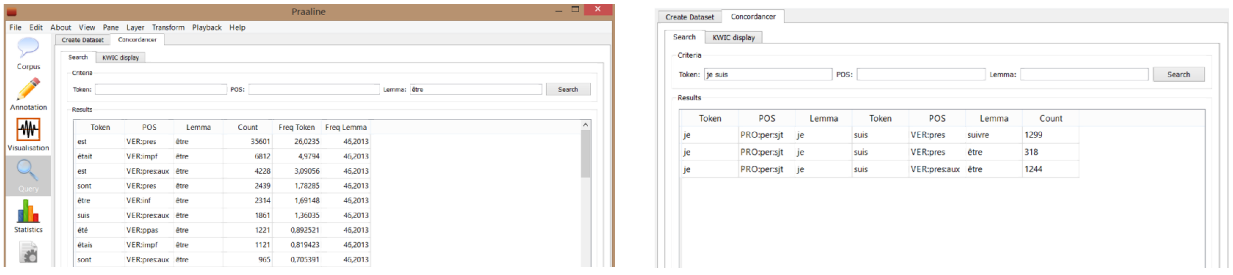


FIGURE 3 : Accès à la concordance par tableau des résultats synthétiques (1) du lemme *être* (gauche), (2) de la chaîne de caractères *je suis* (droite).

Ensuite l'ensemble de ces tableaux donne accès, par des liens hypertextes sur chaque ligne, aux extraits des concordances correspondants. Par exemple, si l'on choisit la deuxième typologie de la cooccurrence *je suis* (lemme *être*, Fig. 3, droite), nous pouvons accéder, entre autres, à la concordance suivante :

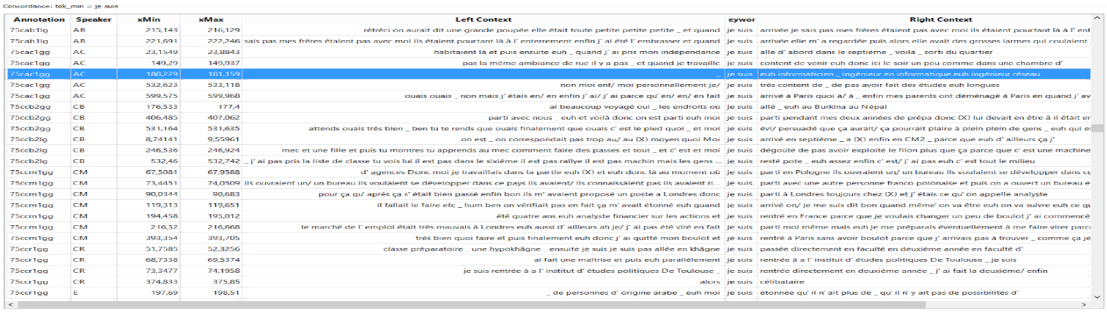


FIGURE 4 : Concordance « *AC : je suis11z euh informaticien. E : Hum hum. AC : Ingénieur en informatique euh ingénieur réseau voilà* » (Corpus PFC_locuteur75cac_discussion guidée).

Enfin un lien multimédia sur chaque pivot (ex. *je suis*) permet de visualiser (à l'aide du logiciel *Praat*, Boersma & Weenink 2014) l'intervalle de la transcription de l'enregistrement contenant les occurrences ou les cooccurrences des mots ciblés, d'accéder aux différentes couches d'annotation et de codage du corpus PFC. Dans l'exemple ci-dessous, plusieurs couches sont associées à la concordance de la séquence *je suis*: (a) transcription de l'intervalle contenant la concordance ciblée ; (b) codage de la *liaison*; (c) codage du *schwa*; (d) étiquettes d'annotation morphosyntaxique.

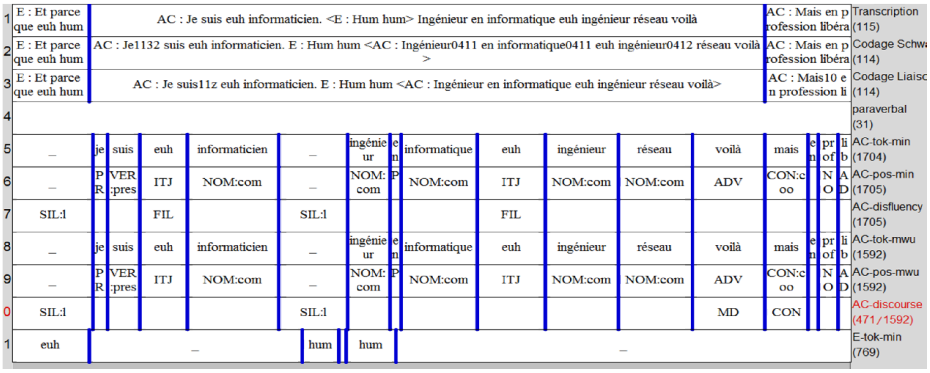


FIGURE 5 : Accès à l'annotation multi-niveaux (au format *TextGrid*) à partir de la concordance.

6 Conclusion et perspectives

Dans cette étude à caractère préliminaire, nous avons présenté les caractéristiques principales d'un prototype de concordancier lemmatisé et multimédia, basé sur le corpus de français oral PFC. Nous cherché à montrer l'intérêt de la création d'un tel concordancier lemmatisé pour le français oral, où les informations linguistiques sont exploitées afin d'améliorer le rappel et la précision des concordances, et de répondre aux besoins de dépouillage précis dans le cadre de la linguistique de corpus. Nous avons utilisé les fonctions traditionnelles des autres concordanciers, tout en proposant un enrichissement des résultats des requêtes grâce à l'apport des données relevant de l'annotation morphosyntaxique du corpus, et du codage de phénomènes phonologiques tels que la liaison et le schwa. De plus, la possibilité d'accéder aux écoutes des intervalles des enregistrements contenant les concordances permet d'élargir les modalités d'accès à l'analyse des contextes d'occurrences des mots ciblés. Dans le cadre de notre travail, nous envisageons d'améliorer l'affichage des annotations supplémentaires dans les concordances détaillées, permettre la sauvegarde des requêtes plus compliquées, d'ajouter une passerelle entre le système de requêtes (concordancier) et le logiciel d'analyse statistique *R*, et éventuellement mettre à les outils développés à la disposition de la communauté (en source libre). L'ensemble de fonctionnalités offertes fait du concordancier un outil fondamental pour l'analyse des corpus linguistiques pour toute discipline des sciences du langage ainsi que pour les domaines de la traduction et de la didactique du FLE.

Références

BERNARD, P., LECOMTE, J., DENDIEN, J., PIERREL, J-M. (2002). Computerized linguistic resources of the research laboratory ATILF for lexical and textualanalysis: Frantext, TLFi, and the software Stella Actes de Language Resources and Evaluation (LREC 2002), 1090-1096, Las Palmas, Espagne.

BOERSMA, P., WEENINK, D. (2014). *Praat: doing phonetics by computer*, ver. 5.3.77, www.praat.org

BOKAN, N. éditeur (2000). *Contemporary Mathematics. Proceedings of the Symposium*, Belgrade.

BYBEE, J. (2001). Frequency effects on French liaison. In (Bybee, Hopper, 2001), 337-359.

BYBEE, J., SCHEIBMAN, J. (1999). The effect of usage on degrees of constituency: the reduction of *don't* in English. *Linguistics* 37(4): 575-596.

BYBEE, J., HOPPER, P., éditeurs (2001). *Frequency and the Emergence of Linguistic Structure*. Amsterdam : John Benjamins.

BYBEE, J., THOMPSON, S. (1997). Three frequency effects in syntax. *Pragmatics and Grammatical Structure* 23: 378-388.

CABALLERO, M. R. (1999). Using a Concordancer in Literary Studies. *TheEuropean English Messenger* 8(2): 59-62. http://www.edict.com.hk/Concordance/

CHRISTODOULIDES, G. (2014). Praaline: Integrating tools for speech corpus research. Actes de *IX Language Resources and Evaluation Conference (LREC 2014)*, 26-31 mai, Reykjiavic, Islande.

CHRISTODOULIDES, G., AVANZI, M., GOLDMAN, J.P. (2014). DisMo: A Morphosyntactic, Disfluency and Multi-Word Unit Annotator. Actes de *IX Language Resources and Evaluation Conference (LREC 2014)*, 26-31 mai, Reykjavik, Islande.

DETEY, S., DURAND, J., LAKS, B., LYCHE, C. (2010). *Les variétés du français parlé dans l'espace francophone: ressources pour l'enseignement.* Paris: Ophrys.

DOSTIE, G. (2004). *Pragmaticalisation et marqueurs discursifs. Analyse sémantique et traitement lexicographique*. Bruxelles: Duculot

DURAND, J., LAKS, B., LYCHE, C. (2002). La phonologie du français contemporain: usages, variétés et structure. In (Pusch et Raible, 2002), 93-106.

GROSS, M. (2000). A bootstrap method for constructing local grammars. In (Bokan, 2000), 229-250.

HEIDEN, S., MAGUÉ, J-P., PINCEMIN, B. (2010). *TXM : Une plateforme logicielle open-source pour la textométrie – conception et développement*. In. I. C. Sergio Bolasco (Ed.), Proc. of 10th International Conference on the Statistical Analysis of Textual Data - JADT 2010) (Vol. 2, p. 1021-1032). Edizioni Universitarie di Lettere Economia Diritto, Roma, Italy.

JACQUET-PFAU, C. (1994). L'intérêt des logiciels de concordances pour la traduction. *Langages* 116: 82-86.

LAKS B. (2005). Phonologie et construction syntaxique: la liaison, un test de figement et de cohésion. *Linx* 53, 155-171.

NEW, B. (2006). Lexique 3 : Une nouvelle base de données lexicales**.** *Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2006),* avril 2006, Louvain, Belgique.

PAUMIER, S. (2003). A Time-Efficient Token Representation for Parsers. Actes de EACL Workshop on Finite-State Methods in Natural LanguageProcessing, 83–90, Budapest, Hongrie.

PINCEMIN, B., ISSAC, F., CHANOVE, M., MATHIEU-COLAS, M. (2006). Concordanciers: Thème et variations, *Lexicometrica*, numéro spécial, 769-780. Actes des Journées d'analyse statistiques des données textuelles (JADT).

PUSCH, C., RAIBLE, W., éditeurs (2002). Romanistische Korpuslinguistik- Korpora und gesprochene Sprache/Romance Corpus Linguistics - Corpora and Spoken Language. Gunter Narr Verlag.

TOGNINI-BONELLI, E. (2001). *Corpus Linguistics at Work*. Amsterdam/Philadelphia : John Benjamins.

TOMASELLO, M. (2000). First steps toward a usage-based theory of language acquisition. *Cognitive Linguistics* 11 : 61-82.

Simulation de l'apprentissage des contextes nominaux/verbaux par n-grammes

Perrine Brusini[1] Pascal Amsili[2] Emmanuel Chemla[3] Anne Christophe[3]
(1) Language, Cognition and Development Lab, Int. School for Advanced Studies (SISSA), Trieste
(2) Laboratoire de Linguistique Formelle, CNRS & Université Paris Diderot
(3) Laboratoire de Sciences Cognitives et Psycholinguistique (CNRS & ENS, EHESS)
contact: pbrusini@gmail.com, amsili@linguist.univ-paris-diderot.fr

Résumé. On présente une étude d'apprentissage visant à montrer que les contextes locaux dans un corpus de parole adressée aux enfants peuvent être exploités, avec des méthodes statistiques simples, pour prédire la catégorie (nominale *vs.* verbale) d'un mot inconnu. Le modèle présenté, basé sur la mémorisation de n-grammes et sur une « graine sémantique » (un petit nombre de noms et verbes supposés connus et catégorisés) montre une excellente précision à toutes les tailles de graine sémantique, et un rappel plus faible, qui croît avec la taille de la graine sémantique. Les contextes les plus utilisés sont ceux qui contiennent des mots fonctionnels. Cette étude de faisabilité démontre que les très jeunes enfants pourraient exploiter les contextes de mots inconnus pour prédire leur catégorie syntaxique.

Abstract. A learning study is presented whose aim is to show that local contexts, in a child-directed speech corpus, can be exploited, with simple statistical methods, to predict the category (noun *vs.* verb) of unknown words. The model we present here is based on the memorisation of n-grams and on a "semantic seed" (a small number of nouns and verbs supposedly known and well categorised). It shows an excellent precision for every size of the semantic seed, and its recall grows along with the size of the semantic seed. The most useful contexts are the ones that include function words. This feasibility study shows that very young children could exploit the contexts of unknown words to predict their syntactic category.

Mots-clés : apprentissage, modélisation de l'acquisition du langage, n-grammes.

Keywords: learning, language acquisition modeling, n-gram.

1 Motivation

On sait que les enfants sont capables de distinguer dès 18 mois les contextes syntaxiques nominaux et verbaux : depuis Shipley *et al.* (1969), de nombreuses expériences comportementales montrent, par exemple, qu'un mot inconnu, associé à une situation où apparaissent à la fois une action nouvelle et un objet nouveau, est associé par les enfants à l'action s'il a été employé dans un contexte verbal (*Regarde, il dase !*) et à l'objet s'il a été employé dans un contexte nominal (*Regarde le dase !*) (Bernal, 2007, par exemple).
Un des aspects qui distinguent les contextes nominaux des contextes verbaux est la nature des mots fonctionnels présents dans ces contextes : le clitique *je* précède exclusivement un verbe (ou d'autres clitiques verbaux), alors qu'un déterminant comme *un* ne précède qu'un nom ou un adjectif. La situation est rendue compliquée, on le sait bien, par la grande quantité de mots fonctionnels homophones (*la* à la fois article (nominal) et clitique (verbal), *son* à la fois adjectif possessif (nominal) et nom plein, *etc.*).
Les mots fonctionnels se distinguent des mots de classe ouverte sur plusieurs plans : ils sont généralement courts et non accentués, ce qui a pu conduire à l'idée qu'ils étaient trop discrets pour être exploités par les enfants en cours d'acquisition (Pinker, 1984), mais ils sont également extrêmement fréquents et situés préférentiellement en bordure d'unité prosodique, ce qui les rend faciles à repérer (Shi *et al.*, 1998). De fait, de très nombreuses études démontrent que les très jeunes enfants reconnaissent les mots grammaticaux de leur langue avant un an (Shi, 2014, pour une revue) et les exploitent pour sélectionner des mots de la catégorie appropriée dès l'âge de 18 mois (Cauvet *et al.*, 2014; Zangl & Fernald, 2007).
L'objectif de l'étude présentée ici est de voir dans quelle mesure les propriétés statistiques de la parole adressée aux enfants peuvent être exploitées par les bébés en phase d'acquisition de leur langue. Il ne s'agit pas d'une hypothèse sur la façon dont le cerveau humain fonctionne, mais d'une investigation de faisabilité, avec quelques hypothèses volontaire-

ment simples et plausibles, correspondant à ce que l'on sait actuellement des capacités linguistiques des jeunes enfants. Il ne s'agit pas non plus de prétendre que d'autres indices, verbaux ou non, présents dans la situation de communication ne sont pas aussi exploités.

On suppose que les enfants, au moment où ils travaillent sur la catégorisation des noms et des verbes (potentiellement dans leur seconde année de vie), possèdent déjà un petit lexique qui peut les aider dans cette tâche — Bergelson & Swingley (2012, 2013) montrent que des enfants de 6 et 9 mois connaissent déjà un certain nombre de noms et verbes. On suppose de plus qu'ils sont capables de regrouper ces mots selon leur catégorie sémantique : par exemple, ils pourraient grouper ensemble *doudou*, *jouet*, et *voiture* parce que ces mots réfèrent à des objets, et *boire*, et *manger* qui réfèrent à des actions [1]. Les noms réfèrant en général à des objets et les verbes à des actions, ces classes sémantiques pourraient servir de point de départ pour la construction des catégories grammaticales nom et verbe.

À partir de cette hypothèse de base, le modèle que nous présentons ici exploite les quelques mots déjà catégorisés pour apprendre les contextes d'occurrence des noms et des verbes, puis utilise cette connaissance pour catégoriser des mots dont la catégorie n'est pas encore connue, sur la seule base de leur contexte.

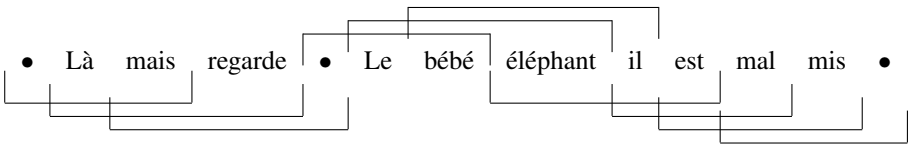
Les hypothèses que nous venons de résumer distinguent de façon assez radicale notre manipulation de la tâche classique de catégorisation morphosyntaxique telle qu'elle est pratiquée en TAL, où la forme de l'item à catégoriser est prise en considération dans le processus, et où l'inventaire des catégories est plus large et plus structuré.

D'autres tentatives de catégorisation non-supervisée sur la base d'indices distributionnels existent dans la littérature (Mintz, 2003; Redington & Finch, 1998). Ces modèles permettent une catégorisation bien meilleure que le hasard, mais souffrent de plusieurs problèmes : soit la catégorisation par le contexte ne fonctionne que sur les mots fréquents (Redington & Finch, 1998), soit elle fonctionne même sur les mots rares ou inconnus, mais crée un grand nombre de classes nominales et verbales distinctes, et surtout ne permet de catégoriser qu'un petit nombre de mots placés dans des contextes très spécifiques (Mintz, 2003). Dans la manipulation présentée ci-après comme dans les tentatives précédentes, le corpus utilisé est segmenté en mots et en tours de parole.

2 Manipulation

Corpus Les corpus sont issus de la base CHILDES 4 (MacWhinney, 2000). Ce sont des transcriptions écrites de paroles spontanées. Notre corpus (133 948 tokens) regroupe différentes transcriptions de deux couples mère-enfant. Nous avons sélectionné les prises de paroles des adultes. Le texte a été catégorisé avec l'étiqueteur de Cordial [2].

Apprentissage Le système construit son modèle en enregistrant les fréquences de tous les n-grammes de mots rencontrés dans le corpus d'apprentissage. Les ponctuations fortes (telles qu'elles sont transcrites dans le corpus) sont traitées comme des mots, mais les n-grammes comprenant une frontière ne sont comptés que si la frontière est le premier ou le dernier élément du n-gramme.



Prédiction Dans la phase de test, les fréquences apprises peuvent être exploitées pour faire une prédiction, à chaque position du corpus de test, sur la base du $(n-1)$-gramme précédent, qu'on appelle le *contexte*. Le mot prédit, dans le contexte $(w_1, \ldots w_{n-1})$, est le mot w tel que le n-gramme formé avec le contexte est le plus fréquent, soit $w = \text{argmax}_w \text{freq}(w_1, \ldots w_{n-1}, w)$. Dans ce cas, le contexte est le contexte immédiatement à gauche de la position-cible. On envisage aussi des prédictions dans des configurations contexte/cible différentes : on parlera de *prédiction droite* pour le cas où le contexte est immédiatement à droite du mot cible, et de *contexte imbriqué* pour le cas où les mots du contexte sont répartis également à droite et à gauche du mot cible (n impair et ≥ 3).

Nous avons décidé de ne pas interpréter les fréquences comme des probabilités, et par conséquent il n'est pas nécessaire de recourir à un lissage au sens strict (de type Good-Turing par exemple), mais nous avons implémenté un repli minimal, pour traiter les cas où le contexte n'a jamais été rencontré dans le corpus d'apprentissage : s'il n'existe pas de mot w tel que le

1. Cette idée est compatible avec des résultats expérimentaux qui établissent que les enfants ont une représentation différente pour les agents et les artéfacts d'un côté et les actions causales de l'autre (Carey, 2009, pour une revue).

2. Étant donné la nature du corpus, le taux de réussite de Cordial n'était pas excellent, même sur les noms et les verbes, où il restait environ 10% d'erreurs de catégorisation (d'après une évaluation faite à la main sur 500 phrases), mais nous avons pu réduire ce taux par un post-traitement semi-automatique.

modèle a rencontré le n-gramme, une prédiction est tentée en prenant comme contexte le $(n-2)$-gramme $(w_2, \ldots w_{n-1})$, et on recommence si nécessaire. Si cette procédure conduit à un contexte nul, le modèle ne fait aucune prédiction.

Projection L'objectif n'est pas à proprement parler de prédire des mots, mais des catégories (plus exactement l'une des deux catégories N ou V). C'est la raison pour laquelle on procède à ce que nous appelons la *projection* du vocabulaire : supposant que le système connaît déjà la nature d'un certain nombre de noms et de verbes, on remplace ces noms et verbes dans le flux d'apprentissage par leur catégorie. Ce faisant, on se distingue des approches classiques en étiquetage morphosyntaxique qui utilisent des chaînes de Markov cachées.

Nous avons sélectionné les noms et les verbes les plus fréquents du corpus comme mots supposés connus, afin de rendre ce processus le plus automatique possible. Comme point de départ, on considère la situation où le système connaît déjà 10% des occurrences de noms et de verbes dans le corpus ce qui correspond à 6 noms et à 2 verbes. Ensuite on a considéré 4 autres « états de vocabulaire » en doublant à chaque fois le nombre de noms et de verbes connus (types), et nous avons enfin considéré le cas où tous les noms et verbes sont connus. La figure 1 illustre l'évolution du corpus d'apprentissage selon le choix de vocabulaire connu.

FIGURE 1 – Illustration de la « projection » du vocabulaire (V_i : 6×2^i N et 2×2^i V connus)

			—	Là	mais	regarde	!	Le	bébé	éléphant	il	est	mal	mis	!
V_0	6 N	2 V	•	Là	mais	regarde	•	Le	N	éléphant	il	est	mal	mis	•
V_1	12 N	4 V	•	Là	mais	V	•	Le	N	éléphant	il	est	mal	mis	•
V_2	24 N	8 V	•	Là	mais	V	•	Le	N	éléphant	il	est	mal	mis	•
V_3	48 N	16 V	•	Là	mais	V	•	Le	N	N	il	est	mal	mis	•
V_4	96 N	32 V	•	Là	mais	V	•	Le	N	N	il	est	mal	V	•
V_m	1310 N	1253 V	•	Là	mais	V	•	Le	N	N	il	est	mal	V	•

Comme on fait l'hypothèse que les mots déjà catégorisés sont ceux dont l'enfant connaît le sens, on peut supposer qu'il identifie ces mots indépendamment de leur forme morphologique [3]. La table 1 donne une idée des mots supposés connus selon la taille du vocabulaire initial.

V_0	6N	doudou	bébé	livre	chose	micro	histoire
	2V	aller			faire		
V_1	V_0+ 6N	pied	poisson	peu [4]	main	lait	nez
	V_0+ 2V	mettre			regarder		
V_2	V_1+12N	caméra	fleur	tête	eau	heure	côté
		oeil	bouche	biberon	assiette	éléphant	fois
	V_1+ 4V	voir			pouvoir		
		dire			falloir		

TABLE 1 – Table présentant quelques-uns des mots connus selon le vocabulaire initial

Test Le test est pratiqué sur un extrait nouveau du corpus, et consiste à comparer la prédiction faite par le modèle avec le mot/la catégorie effectivement présente, et ce pour des *positions cible* particulières, qui doivent vérifier les deux conditions suivantes : (1) le mot du contexte le plus proche de la cible doit avoir été rencontré dans le corpus d'apprentissage (ce qui revient à ne pas faire de repli jusqu'au contexte nul) [5] ; (2) la cible à catégoriser doit être peu fréquente dans le corpus. Nous avons choisi le seuil de 0,05% [6].

Mesures Le parti pris de faire l'apprentissage sur un flux unique dans lequel apparaissent aussi bien des mots que des catégories conduit à des réponses diverses de la part du système : il peut aussi bien prédire une catégorie N ou V qu'un mot quelconque, qui peut être un nom ou un verbe n'appartenant pas au vocabulaire connu.
Les réponses du système sont classées en trois catégories : N, V, ni N ni V ; et on définit une mesure de rappel et une mesure de précision pour les deux catégories N et V. On compte une *bonne réponse 'nom'* (BR_N) si le système a répondu N lorsqu'il y avait un nom dans le corpus de test. Ceci signifie que toutes les réponses où le modèle donne un nom spécifique (comme *girafe* ou *poupée*) sont comptées comme un *manqué* (MA_N), comme les cas où une autre réponse que

3. Pour les noms cette hypothèse est validée d'office puisque dans leur grande majorité ils ne distinguent pas à l'oral le singulier du pluriel ; et pour la plupart des verbes les formes fléchies varient peu dans notre corpus (les auxilliaires *être* et *avoir* ont été exclus).
4. *Peu* est un adverbe mais il a des emplois en fonction nominale.
5. Cette condition élimine environ 4% d'occurrences de noms et 9% d'occurrences de verbes.
6. Ce seuil garantit qu'on ne tente pas de prédiction sur les mots déjà connus ; il a de plus la propriété d'éliminer la quasi-totalité des mots fonctionnels, qui sont très fréquents, alors que la majorité des mots de classe ouverte sont moins fréquents que ce seuil (Brusini, 2012, p. 167).

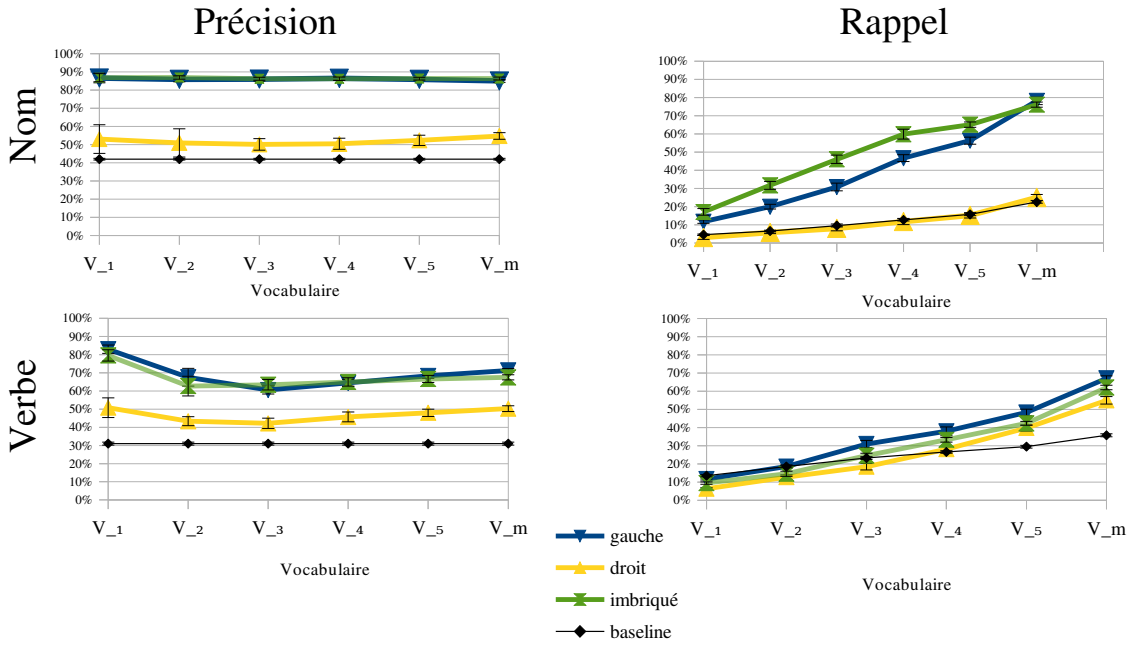
N est donnée. Si le système répond N alors qu'il n'y a pas de nom dans le corpus de test, on compte une *fausse alarme 'nom'* (FA_N). Les mêmes calculs sont effectués pour les réponses V. On peut alors définir de façon classique une précision et un rappel pour chaque catégorie X : $pre_X = \frac{BR_X}{BR_X+FA_X}$ $rap_X = \frac{BR_X}{BR_X+MA_X}$

Baseline Nous prenons comme baseline un modèle qui prédit N, V, ni N ni V, en fonction de la distribution de ces trois "catégories" dans le corpus d'apprentissage. Les distributions varient selon le vocabulaire connu.

Validation croisée Afin de déterminer l'influence de la taille du corpus d'apprentissage sur les performances du modèle et pour pouvoir tester la reproductibilité des résultats obtenus, nous avons procédé à une validation croisée : le corpus a été découpé en 10 sous-corpus de taille égale, et dans chaque cas 2/3 du corpus a servi pour l'entraînement, et 1/3 pour le test. Cette manipulation permet d'une part de calculer un écart-type des modèles (et donc d'évaluer la reproductibilité des résultats) et permet d'autre part d'évaluer l'impact de la taille du corpus d'apprentissage sur la performance. Les barres d'erreurs de la figure 2 correspondent à cet écart-type.

3 Résultats

FIGURE 2 – Performance des 3 modèles en précision et en rappel pour les catégories N et V



Les résultats des calculs pour $n = 3$, pour les contextes gauches, droits et imbriqués, en précision et rappel, comparés à la baseline, sont donnés à la figure 2. On note tout d'abord que les 3 variantes qui exploitent le contexte ont de meilleures performances que la baseline [7]. On peut donc conclure qu'il y a de l'information dans les contextes d'occurrence de noms et de verbes qui permet de catégoriser des mots nouveaux. Le modèle utilisant le contexte droit (en jaune) est le modèle le moins efficace dans cette tâche de catégorisation, sa précision et son rappel sont plus bas que les modèles gauche (bleu) et imbriqué (vert) pour les noms comme pour les verbes [8]. Les deux autres modèles sont très efficaces tous les deux, avec un léger avantage pour le modèle utilisant les contextes imbriqués, notamment pour le rappel des noms où il est significativement meilleur que le modèle gauche [9]. De manière générale, les résultats sont meilleurs pour N que pour V.

Il est frappant de constater que la précision des différents modèles ne s'améliore pas lorsqu'on augmente le nombre de noms et de verbes connus lors de l'entraînement. Même à très petit vocabulaire, la précision des deux meilleurs modèles est excellente, autour de 90% pour les noms et 70% pour les verbes. Au contraire, le rappel augmente avec la taille de

7. Pour l'ensemble des comparaisons entre les 3 modèles et la baseline : Précision : $F(1,9) > 40$; $p < 0.001$; Rappel : $F(1,9) > 13$; $p < 0.01$.
8. Précision : $F(1,9) > 180$; $p < 1.0\times10^{-6}$ pour l'ensemble des comparaisons entre les modèles gauche et imbriqué avec le modèle droit. Rappel : ensemble des comparaisons : $F(1,9) > 50$; $p < 1.0\times10^{-4}$.
9. Comparaison des 2 modèles : $F(1,9) = 10,6$; $p = 0.01$ pour la précision et $F(1,9) = 218,6$; $p = 1.3\times10^{-7}$ pour le rappel.

vocabulaire [10] : en partant d'une performance faible à petit vocabulaire (moins de 20% des noms et verbes identifiés), on arrive à une performance plus acceptable à vocabulaire moyen (avec 50% des verbes bien catégorisés lorsque 48 noms et 16 verbes sont connus (V_3), et 50% des noms à 24 noms et 8 verbes connus (V_2)).

On peut aussi constater que la variabilité est très faible, les barres d'erreurs étant à peine visibles parfois. Cela signifie que les performances de nos modèles sont très stables, alors qu'ils sont entraînés et testés sur des parties différentes du corpus. Le même test effectué sur le corpus entier (dix fois plus grand) fournit des résultats très similaires, ce qui indique que la taille du corpus d'apprentissage n'influe pas sur les performances du modèle.

4 Discussion

Le modèle présenté ici a pour particularité de présupposer des connaissances initiales plausibles — la graine sémantique — et de faire appel à des calculs très simples, puisqu'il se contente de comptabiliser la fréquence d'apparition des contextes et sélectionne l'élément le plus souvent rencontré dans un contexte donné. En dépit de cette simplicité, le modèle obtient une excellente performance en termes de précision, et ce même pour la plus petite taille de graine sémantique, seulement 6 noms et 2 verbes supposés connus — ce résultat est vrai pour les contextes gauche et imbriqué. Autrement dit, lorsque le modèle catégorise un mot, il le fait en général de manière correcte (80% pour les noms, 70% pour les verbes). Cette expérience montre que les contextes immédiats des noms et des verbes contiennent beaucoup d'information sur leur catégorie syntaxique.

Ce résultat est d'autant plus surprenant que le modèle opère sa catégorisation en regardant uniquement le contexte : le mot-cible lui-même n'est pas pris en compte lors de l'étape de catégorisation. Cet aspect du modèle est particulièrement désirable lors de l'acquisition d'un lexique, pour deux raisons principales. Premièrement, le modèle est tout à fait capable de catégoriser un mot-cible qu'il n'a encore jamais rencontré, pour peu qu'il soit entendu dans un contexte connu : en termes d'acquisition, un enfant qui rencontre un mot pour la première fois pourrait calculer sa catégorie syntaxique, nom ou verbe, et l'exploiter pour contraindre le sens de ce mot — si c'est un nom il réfère probablement à un objet, si c'est un verbe, à une action. Cette stratégie semble être utilisée par les enfants (Bernal, 2007) par contraste avec la stratégie de (Redington & Finch, 1998), où ce sont les mots connus qui sont catégorisés. Deuxièmement, le lexique contient de nombreux homophones qui appartiennent à des catégories syntaxiques différentes, comme *la ferme*/*je ferme*. Du fait que le modèle catégorise les mots grâce à leur seul contexte, il n'est pas gêné par de tels homophones, qui sont pour lui des mots comme les autres. Au contraire, un modèle (ou un enfant) qui entreprendrait de classifier les mots en prenant en compte à la fois le contexte et la catégorie supposée du mot en cours de classification (telle que stockée dans le lexique), devrait éprouver des difficultés à apprendre le deuxième membre d'une paire d'homophones lorsque le premier membre est déjà connu.

L'excellente performance observée en précision, indépendamment de la taille de la graine sémantique, est en contraste frappant avec la performance en rappel : celui-ci est faible à la plus petite taille de graine sémantique, et augmente au fur et à mesure que la graine sémantique augmente. Intuitivement, lorsque le modèle connaît initialement peu de mots (petite graine sémantique), alors il apprend un petit nombre de contextes de noms et de verbes : lorsqu'il utilise ces contextes pour catégoriser un nom ou un verbe, sa performance est bonne (précision élevée), mais il « rate » beaucoup de noms et de verbes qui apparaissent dans des contextes non reconnus comme prédicteurs de noms ou de verbes. Cette analyse intuitive est confirmée par l'examen des manqués du modèle : pour les petites tailles de graine sémantique, les manqués correspondent le plus fréquemment à des cas où le modèle répond un nom spécifique au lieu de répondre avec la catégorie N (resp. V). En ce qui concerne l'acquisition, ce comportement ne pose pas de problème particulier : en effet, en l'absence d'information pertinente dans un contexte donné, l'enfant peut attendre de rencontrer le mot dans un contexte plus informatif. Au contraire, il serait désastreux de multiplier les fausses alarmes : si l'enfant pense qu'un mot donné est un verbe et réfère à une action, alors qu'en réalité c'est un nom qui réfère à un objet, il apprendra un mauvais sens. Autrement dit, il est préférable pour un apprenant d'avoir une bonne précision et un mauvais rappel, et c'est ce que fait le modèle aux petites tailles de graine sémantique.

L'examen des contextes utilisés le plus fréquemment pour la catégorisation montre qu'ils contiennent majoritairement des mots grammaticaux. Pour les contextes de nom, le mot-cible est le plus souvent immédiatement précédé d'un article ; pour les contextes de verbe, il s'agit de pronoms personnels ou relatifs, et d'auxiliaires. Ainsi, bien que le modèle ne contienne aucune hypothèse *a priori* sur l'utilité potentielle des mots grammaticaux, ceux-ci émergent spontanément dans les contextes utiles, du simple fait de leur fréquence et de leur distribution.

10. Effet de Vocabulaire pour chaque modèle : N : $F(6,54) > 50$; $p < 1.0\times10^{-15}$; V : $F(6,54) > 200$; $p < 1.0\times10^{-15}$.

La comparaison des modèles à contexte gauche, imbriqué et droit montre une performance bien pire sur le contexte droit (il faut noter que le modèle imbriqué se replie sur l'élément de gauche lorsque le contexte complet n'est pas connu, ce qui fait qu'il y a un recouvrement entre les modèles gauche et imbriqué). Ceci pourrait être une conséquence du fait que le français est récursif à droite, avec en général les mots grammaticaux à gauche des têtes lexicales. Une autre possibilité serait que comme les mots de gauche sont entendus avant le mot-cible, et ceux de droite après, les langues s'organisent de manière à maximiser la quantité d'information disponible avant le mot lui-même. Pour trancher entre ces deux hypothèses, il faudrait effectuer ces mêmes calculs avec une langue récursive à gauche, comme le japonais.

Cette étude démontre la pertinence d'une approche de simulation dont les hypothèses sont contraintes par les résultats d'expérimentation psycholinguistique avec les très jeunes enfants. Par exemple, ce modèle démontre que l'utilisation des mots fonctionnels comme prédicteurs de catégorie pour les mots de contenu ne semble pas nécessiter de construire *a priori* des catégories de mots fonctionnels, telles que *déterminant* ou *clitique sujet* : la simple reconnaissance de l'item pourrait suffire. Ce résultat est très intéressant car l'homophonie entre mots fonctionnels rend leur catégorisation difficile. En retour, les résultats de modèles de simulation comme celui qui est présenté ici pourra permettre de faire des prédictions testables expérimentalement chez les très jeunes enfants, pour peut-être un jour parvenir à un modèle computationnel psychologiquement plausible de l'acquisition des catégories syntaxiques par les jeunes enfants.

Remerciements

Ce travail a bénéficié du soutien de l'Agence Nationale de la Recherche (grants n° ANR-2010-BLAN-1901, ANR-13-APPR-0012, ANR-10-IDEX-0001-02 PSL* and ANR-10-LABX-0087 IEC) et de la fondation de France. Nous remercions Benoît Crabbé pour son aide dans la réalisation de la première version du modèle.

Références

BERGELSON E. & SWINGLEY D. (2012). At 6-9 months, human infants know the meanings of many common nouns. *Proceedings of the National Academy of Sciences of the United States of America*, **109**(9), 3253–3258. doi:10.1073/pnas.1113380109.

BERGELSON E. & SWINGLEY D. (2013). The acquisition of abstract words by young infants. *Cognition*, **127**(3), 391–397. doi:10.1016/j.cognition.2013.02.011.

BERNAL S. (2007). *De l'arbre (syntaxique) au fruit (du sens) : Interactions des acquisitions lexicale et syntaxique chez l'enfant de moins de 2 ans*. PhD thesis, Université Pierre et Marie Curie.

BRUSINI P. (2012). *Découvrir les noms et les verbes : Quand les classes sémantiques initialisent les catégories syntaxiques*. PhD thesis, Université Pierre et Marie Curie.

CAREY S. (2009). *The origin of concepts*. Oxford University Press.

CAUVET E., LIMISSURI R., MILLOTTE S., SKORUPPA K., CABROL D. & CHRISTOPHE A. (2014). Syntactic context constrains lexical access in French 18-month-olds. *Language Learning and Development*, **10**(1), 1–18.

MACWHINNEY B. (2000). *The CHILDES Project : Tools for analyzing talk*. Mahwah, NJ : Lawrence Erlbaum Associates. Third Edition.

MINTZ T. H. (2003). Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, **90**(1), 91–117.

PINKER S. (1984). *Language Learnability and Language Development*. Cambridge, MA : Harvard University Press.

REDINGTON M. N. C. & FINCH S. (1998). Distributional information : A powerful cue for acquiring syntactic categories. *Cognitive Science*, **22**(425–469).

SHI R. (2014). Functional morphemes and early language acquisition. *Child Development Perspectives*, **8**(1), 6–11.

SHI R., MORGAN J. L. & ALLOPENNA P. (1998). Phonological and acoustic bases for earliest grammatical category assignment : a cross-linguistic perspective. *Journal of Child Language*, **25**, 169–201.

SHIPLEY E. F., SMITH C. S. & GLEITMAN L. R. (1969). A study in the acquisition of language : Free responses to commands. *Language*, **45**(2), 322–342.

ZANGL R. & FERNALD A. (2007). Increasing flexibility in children's online processing of grammatical and nonce determiners in fluent speech. *Language Learning and Development*, **3**(3), 199–231.

Impact de la nature et de la taille des corpus d'apprentissage sur les performances dans la détection automatique des entités nommées

Ollagnier Anaïs[1, 2], Fournier Sébastien[1], Bellot Patrice[1,2], Béchet Frédéric[3]
(1) Aix-Marseille Université, CNRS, ENSAM, Université de Toulon LSIS UMR 7296, 13397, Marseille, France
(2) Aix-Marseille Université, CNRS, CLEO OpenEdition UMS 3287, 13451, Marseille, France
(3) Aix-Marseille Université, CNRS, LIF UMR 7279, 13288, Marseille, France
anais.ollagnier@openedition.org

Résumé. Nous présentons une étude comparative sur l'impact de la nature et de la taille des corpus d'apprentissage sur les performances dans la détection automatique des entités nommées. Cette évaluation se présente sous la forme de multiples modulations de trois corpus français. Deux des corpus sont issus du catalogue des ressources linguistiques de l'Association Européenne pour les Ressources Linguistiques (ELRA) et le troisième est composé de documents extraits de la plateforme OpenEdition.org.

Abstract. We present a comparative study on the impact of the nature and size of the training corpus on performance in automatic named entities recognition. This evaluation is in the form of multiple modulations on three French corpus. Two corpora are from the catalog of the European Language Resources Association (ELRA) and the third is composed of documents extract from the OpenEdition.org platform.

Mots-clés : Reconnaissance d'entités nommées, Adaptation au domaine, comparaison d'outils.

Keywords: Named entity recognition, Domain adptation, performance comparison.

1 Introduction

La reconnaissance des entités nommées (REN) est une sous tâche de l'activité d'extraction d'information. Elle consiste à rechercher des objets textuels (c'est à dire un mot ou un groupe de mots) catégorisables dans des classes telles que les noms de personnes, les noms d'organisations ou d'entreprises, les noms de lieux, etc. De nombreuses campagnes d'évaluation, telles que la Message Understanting Conference (Grishman & Sundheim, 1996), l'Automatic Content Extraction [1] ou encore la Document Understanding Conference [2] pour l'Europe, permettent d'évaluer et de comparer les différents systèmes de détection d'entités nommées. Les résultats présentés lors des grandes campagnes d'évaluation montrent régulièrement des F-mesures avoisinant les 90% (Marsh & Perzanowski, 1998). Cependant ces résultats sont à prendre avec du recul, en effet, les tâches d'évaluation proposées sont souvent spécialisées et orientées sur un seul type de données, et les capacités de généralisation des systèmes sont rarement évaluées. Cependant les besoins en annotation en entités sont très divers, que ce soit par rapport au jeu d'étiquettes considéré, au type de texte ou au domaine sémantique visés. Il est donc intéressant d'évaluer les capacités de généralisation des systèmes de REN ainsi que d'étudier des processus d'adaptation peu couteux à un nouveau cadre d'utilisation.

Dans cet article nous allons nous focaliser sur l'étude de cette capacité à généraliser pour un système donné à travers le traitement de corpus provenant de la plateforme OpenEdition [3]. Ces corpus sont particulièrement intéressants car ils présentent une diversité de formes (livres, revues, actualité et blog) ainsi que de contenu (thèmes très variés issus des domaines des SHS). En confrontant des outils génériques d'extraction d'entités nommées à ces corpus, nous allons pouvoir déterminer quelle est la meilleure stratégie pour adapter un système existant à un nouveau cadre d'utilisation, tout en minimisant l'effort d'adaptation.

1. http ://www.itl.nist.gov/iad/mig/tests/ace/
2. http ://duc.nist.gov/
3. http ://www.openedition.org/

2 Détecteur d'entités nommées et corpus d'apprentissage

Les travaux en REN sont classés principalement selon trois types d'approches : les approches symboliques, les approches statistiques et les approches hybrides. Les approches symboliques (Poibeau, 2003) se basent sur des règles écrites à la main, ces approches ont l'avantage de privilégier la précision des détections mais présentent l'inconvénient d'être très dépendantes du domaine d'application et d'être coûteuses à mettre en place si l'on veut obtenir une couverture satisfaisante (Nadeau & Sekine, 2007). Les approches statistiques (Burger *et al.*, 2002) se basent sur un mécanisme d'apprentissage à partir d'un corpus pré-étiqueté. De manière générale, ce type d'approche privilégie le rappel plutôt que la précision. Cependant, comme précisé précédemment, ces approches nécessitent l'utilisation de données annotées comme base d'apprentissage. Les approches hybrides consistent en une combinaison des deux approches précédentes (Shaalan & Oudah, 2014). Les campagnes d'évaluation des systèmes de REN ont montrées que les différences de performance entre les systèmes dépendent plus du soin mis dans l'adaptation fine des modèles (règles ou traits) que des choix des modèles théoriques. Ainsi les systèmes à base de règles rédigées manuellement (soit purement symboliques, soit hybrides) sont souvent les plus performants (Sun, 2010) lors des conférences ACL, MUC-7 ou CoNLL03. Cependant ces résultats sont constatés sur des domaines précis, lors d'évaluations sur des domaines plus larges les résultats se dégradent (Mansouri *et al.*, 2008). Le but de cette étude étant d'adapter à moindre coût des systèmes de REN à de nouveaux types de corpus, nous avons opté pour l'utilisation d'outils basés sur des approches statistiques pour lesquelles nous avons pu trouver des corpus français déjà annotés. Afin d'établir nos corpus d'apprentissage nous avons utilisé deux corpus déjà annotés en EN en français extraits du catalogue ELRA [4] : le corpus Quaero d'émissions télé-radio-diffusées (ELRA-S0349) et le corpus Quaero de presse anciennes (ELRA-W0073). Ainsi qu'un troisième corpus constituant la *cible* de notre étude a été annoté manuellement à partir des différents types de documents présents sur la plateforme *OpenEdition.org*. Dans les sections suivantes nous allons détailler les différentes caractéristiques de ces corpus ainsi que les outils de détection des entités nommées choisis.

2.1 Corpus Open Edition (OE)

Un corpus issu de la plateforme *OpenEdition.org* a été extrait et entièrement annoté manuellement par un seul annotateur. Il se compose de documents extraits des quatre plateformes Open Edition (Revues, Books, Calenda, Hypothèses) qui est un vaste catalogue de publications scientifiques en sciences humaines et sociales (SHS). La première plateforme, *Revues.org* est composée de revues en SHS. La seconde, *Books* met à disposition en libre accès l'intégralité du contenu de plus de mille ouvrages en SHS. La troisième plateforme, *Calenda* répertorie des actualités dans le domaine des SHS. Enfin la quatrième plateforme, *Hypothèses* est constituée de blogs académiques. Ces différentes plateformes dédiées à la valorisation des publications dans les sciences humaines et sociales possèdent plusieurs spécificités que ce soit au niveau structurel que thématique. Parmi ces spécificités nous pouvons citer : des documents de longueur variable, des dimensions stylistiques variées, la présence de multilinguisme ainsi que l'utilisation pour certains documents (notamment au sein des plateformes Books et Revues) de guide de recommandation aux auteurs. La table 1 présente des exemples de phrases annotés en entités nommées du corpus OE.

2.2 Corpus Quaero

L'ELRA met à disposition deux corpus Quaero. Ces corpus ont été entièrement annotés manuellement selon la définition étendue et structurée d'entités nommées Quaero, qui distingue les "types" et les "composants" d'entités (Rosset *et al.*, 2011). Dans le cadre de notre évaluation, nous avons décidé de conserver les annotations les plus communes entre nos trois corpus, à savoir, les noms de personnes, les toponymes et les noms d'organisations. Les entités structurées de Quaero ont été "*aplaties*" en ne gardant que la structuration de plus haut niveau (Ex : <pers.ind> <name.first> Jacques </name.first> <name.last> Chirac </name.last> </pers.ind> : <pers> Jacques Chirac </pers>). Deux corpus ont été utilisés :

1. Corpus Quaero d'émissions télé-radio-diffusées annoté en entités nommées (QO)
   Le corpus QO consiste en l'annotation manuelle du corpus ESTER2 et du corpus d'évaluation de systèmes de reconnaissance de la parole Quaero (Galliano *et al.*, 2006). Ce corpus comporte la particularité de n'être

4. http ://www.elra.info/ELRA.html

Books Plus tard, <pers> R. Janko </pers> s'est inspiré des remarques de <pers> T. Weischadle </pers>. À propos de l'introduction, il relève plusieurs exceptions aux remarques formulées par <pers> T. Weischadle </pers>, parmi lesquelles la présence du nom du dieu au vocatif.
Revues <pers> Michael Spens </pers>, (<pers> M. Spens </pers>, Paysages contemporains, <date> 2005 </date>) examine les créations des paysagistes de différentes générations <pers> B. Lassus </pers>, <pers> L. Halprin </pers>
Blog C'est <pers> Jean-Marc Berlière </pers> qui l'annonce: les <org> Archives de la Préfecture de Police </org> vont fermer pour déménager au <loc> Pré Saint-Gervais </loc>, dans la banlieue est de la capitale.
Calenda <date> 12 février </date> (9h30-12h) - Collecter - « Les éboueurs entre collecte des ordures ménagères et récupération », <pers> Christophe Clerfeuille </pers> (éboueur, président de l'<org> association Collectif Ripeurs </org>) et <pers> Stéphane Le Lay </pers> (sociologue, <org> CNAM </org>)

Table 1 – Exemples de phrases annotées en EN du corpus OE provenant du site *OpenEdition.org*

constitué que de données non structurées. En outre, les textes comportent de nombreuses disfluences de surface (hésitations, répétition, etc.) qui peuvent générer des difficultés dans la généralisation des modèles. Ce corpus a été utilisé lors de la campagne ESTER2 [5] durant laquelle les meilleurs résultats oscillent entre 78,4% et 92,5% de F-mesure (Nouvel *et al.*, 2010). Dans le cadre de notre évaluation, nous avons utilisé la partie dédiée à l'apprentissage constituée d'actualités télé-radio-diffusées, soit 188 émissions pour 1 291 225 mots afin de constituer nos corpus d'apprentissage.

2. Corpus Quaero de presse ancienne étendue en entités nommées (QP)
   Le corpus QP consiste en l'océrisation de 76 numéros de journaux, publiés entre 1890-1891, fournis par la Bibliothèque Nationale de France. Trois publications sont utilisées (Le Temps, La Croix et Le Figaro) pour un total de 295 pages. Ce corpus présente plusieurs caractéristiques. Premièrement, il est entièrement constitué de documents OCR-isés dont le taux de qualité a été estimé bon par rapport à l'état de l'art dans le domaine (Character Error Rate de 5,09 et Word Error Rate de 36,59) (Rosset *et al.*, 2012). Quelques erreurs résiduelles persistent notamment dans la reconnaissance de certain caractère comme le « e » souvent écrit « o ». Deuxièmement, ce corpus réfère à une période assez ancienne dont les informations diffèrent des actuelles. Et troisièmement, la particularité des journaux OCR-isés basés sur des éditions papiers se retrouve également dans leurs structures en colonnes. De ce fait, les textes conservent de nombreux sauts de ligne ainsi que de nombreuses césures. Dans l'article de Rosset (Rosset *et al.*, 2012), trois systèmes de REN à base de méthodes statistiques sont évalués sur ce corpus. Les résultats présentent un taux d'erreur oscillant entre 44,2 et 60,3 (Slot Error Rate). Dans le cadre de notre évaluation, nous avons utilisé la partie dédiée à l'entraînement, constituée de 231 pages pour 1 297 742 mots afin de constituer nos corpus d'apprentissage.

2.3 Les outils

Nous avons choisi plusieurs outils basés sur des approches d'apprentissage statistiques, à savoir, l'entropie maximale (Chieu & Ng, 2002) ainsi que les champs conditionnels aléatoires (CRFs) (Sobhana *et al.*, 2010), avec et sans *boosting* (Favre *et al.*, 2005). Le premier outil que nous avons testé est intégré dans l'environnement OpenNLP [6]. Plusieurs outils sont disponibles au sein de cette plateforme qui s'intéresse à la tokénisation, la détection de syntagmes, la détection des entités nommées (NameFinder), tous basés sur le *framework* Maxent implémentant des modèles à maximum d'entropie. Le second est Stanford NER [7], il fournit un détecteur d'entités nommées basé sur des CRFs. Une évaluation de ces deux outils présentée par (Rodriquez *et al.*, 2012) sur des documents OCR-isés anglais extraits du Wiener Library dataset et du King College London dataset montre une meilleure performance globale pour Stanford NER (F-mesure : 54%) et une performance beaucoup plus faible pour OpenNLP surtout au niveau de la précision (F-mesure : 19%). Le troisième outil que nous avons choisi d'utiliser est LiaNE [8], basé sur une combinaison d'un modèle HMM pour la prédiction d'étiquettes pour chaque

5. http ://www.afcp-parole.org/camp_eval_systemes _transcription/index.html
6. http ://opennlp.apache.org
7. http ://nlp.stanford.edu/software/CRF-NER.shtml
8. http ://pageperso.lif.univ-mrs.fr/ frederic.bechet/download.html

mot et d'un modèle CRF pour l'étiquetage de séquences constituant les entités nommées. Sa particularité est d'être orienté sur le traitement de données orales. LiaNE a obtenu lors de l'évaluation ESTER2 une F-mesure de 78,4% (Nouvel *et al.*, 2010).

3 Expérimentations

Notre évaluation s'est déroulée en deux parties. La première partie s'est établie en tenant compte de la nature des corpus, c'est à dire que nous avons voulu au vu des spécificités des deux corpus Quaero, évaluer l'impact de leurs caractéristiques lors de l'apprentissage des outils ainsi que leur portabilité. La seconde partie de notre évaluation s'est focalisée sur l'adaptation des modèles en utilisant une partie du corpus OE pour apprendre ou adapter les modèles d'EN de nos systèmes.

Evaluation de la nature et de la taille des corpus d'apprentissage Dans le cadre de nos évaluations de l'impact de la nature et de la taille des corpus d'apprentissage sur les performances en NER, nous avons utilisé les données du corpus OE afin de constituer notre corpus de test et les corpus QO et QP comme corpus d'apprentissage. Les trois systèmes LiaNE, OpenNLP et StandfordNER sont comparés dans la table 2 et la table 3.

	Précision (%)		Rappel (%)		F-mesure (%)	
Corpus	QO	QP	QO	QP	QO	QP
LiaNE	**55,7**	15,5	48,3	21,9	**51,8**	18,1
Open NLP	30,4	27,3	29,0	23,7	29,7	25,4
Stanford NER	44,0	**36,4**	**52,3**	**50,7**	47,8	**42,3**

Table 2 – Performance en NER sur le corpus de test OE avec 3 systèmes et 2 corpus d'apprentissage différents

De manière générale, nous pouvons constater que les performances sont beaucoup plus faibles que celles obtenues par ces mêmes systèmes dans la littérature. Cette dégradation a deux explications : d'une part à cause des spécificités du corpus OE, assez différent des corpus Quaero utilisés pour l'apprentissage ; d'autre part à cause de l'utilisation directe des corpus Quaero pour l'apprentissage des systèmes, sans adaptation des traits utilisés pour décrire les données d'apprentissage. Cette table nous permet de constater un certain écart sur les performances globales des outils sur les différents corpus. Nous pouvons noter une constante assez faible de la part de OpenNLP. A l'inverse, nous pouvons constater que LiaNE obtient une performance bien meilleure après son entraînement sur le corpus QO, ce qui s'explique par la participation de LiaNE à la campagne ESTER2, ce qui a favorisé son adaptation aux données non structurées présentes dans le corpus QO. Les résultats obtenus par Stanford NER sont les plus constants. Au vu de cette table nous pouvons tirer deux enseignements de ces résultats : premièrement, malgré le fait que le corpus QP contienne du texte écrit *a priori* plus proche des données OE que le corpus oral QO, les résultats obtenus en entraînant les systèmes sur QP sont globalement plus faible que ceux obtenus avec QO. Sans doute la distance temporelle entre le données QP et OE est-elle trop forte. Deuxièmement, en dehors de la méthode et du choix du corpus d'apprentissage, les performances des systèmes sont fortement affectées par l'adaptation des traits utilisés pour décrire les données. Ainsi, bien que les deux systèmes LiaNE et Standford NER partagent tous deux une approche à base de CRF, le système LiaNE donne les meilleurs résultats sur le corpus QO alors qu'il se dégrade sur le corpus QP ; tandis que le système Standford NER affiche plus de stabilité sur les deux corpus, bien qu'atteignant des performances plus faibles. On peut donc voir que l'adaptation spécifique du système LiaNE sur les données QO lui permet de tirer un meilleur parti de ces données lors de l'apprentissage.

Outils	F-mesure (%)				
Partitions	100%	80%	60%	40%	20%
LiaNE	51,8	52,1	**52,5**	39,3	37,0
Open NLP	29,7	29,8	**30,1**	27,1	26,9
Stanford NER	47,8	48,9	**50,6**	45,8	44,7

Table 3 – Les valeurs de F-mesure obtenues sur le corpus OE en faisant varier la taille du corpus d'apprentissage QO pour chacun des outils

La deuxième expérience effectuée sur les mêmes données a consisté à faire varier la taille du corpus d'apprentissage QO. Plusieurs corpus ont été constitués en découpant proportionnellement le corpus QO, soit 1 291 225 mots. Une scission de 20% composée de 433 951 mots, une scission de 40% de 657 071 mots, une scission de 60% de 888 280 mots et une scission de 80% de 1 000 009 mots. Nous avons ensuite évalué ces données d'apprentissage en les testant sur le corpus OE soit 102 744 mots. Les résultats présentés dans la table 3 nous permettent de constater que, de manière générale, la quantité de données d'apprentissage n'a pas un impact linéaire sur les performances des systèmes. Nous pouvons constater que pour les trois outils les meilleurs résultats sont obtenus sur la scission constituée de 60% du corpus d'apprentissage. Plus précisément, nous pouvons noter une différence moyenne de 1,3% entre la scission de 60% et 100% et une différence moyenne de 8,4% entre la scission de 60% et 20%. Ces constations nous permettent de corroborer plusieurs études qui démontrent, que pour certaine tâche, un trop grand nombre de données d'apprentissage ne correspondant pas exactement aux données de tests ne permet pas d'augmenter la couverture mais qu'au contraire ceci peut engendrer du bruit et de la confusion (Biber, 1993; Gildea, 2001).

Evaluation de l'adaptation des modèles au corpus OE Cette série d'évaluation a pour but de mesurer l'impact sur les performances de nos modèles de l'ajout de données correspondant à la tâche cible. Pour cela nous présentons deux séries d'expériences effectuées en validation croisée sur le corpus OE et OE+QO. Chaque corpus d'apprentissage de OE (5 validations croisées) est composé d'environ 79 000 mots pour environ 1200 noms de personnes, 495 toponymes et 678 noms d'organisations. Les corpus de test sont composés d'environ 22 700 mots pour environ 560 noms de personnes, 220 toponymes et 205 noms d'organisations. Les évaluations OE+QO ajoutent le corpus QO à chaque partition d'apprentissage du corpus OE.

	Précision (%)		Rappel (%)		F-mesure (%)	
Corpus	OE	OE+QO	OE	OE+QO	OE	OE+QO
LiaNE	18,1	**60,0**	**66,7**	**56,3**	28,6	**57,9**
Open NLP	20,0	28,7	52,8	42,9	28,3	34,3
Stanford NER	**44,3**	56,3	56,3	48,3	**49,1**	51,1

Table 4 – Performances obtenues en évaluation croisée sur le corpus OE avec ou sans le corpus QO lors de l'apprentissage

Nous pouvons voir que malgré la petite taille du corpus OE, comparé au corpus QO, le système Standford NER permet d'obtenir des performances comparables (F-mesure : 49,1%) aux meilleures performances obtenues dans les tables 2 et 3. Le système LiaNE, lorsqu'il est appris uniquement sur le corpus OE présente de bons résultats de rappel, mais la présence de trop nombreux faux positifs conduit à un effondrement de la précision. L'ajout du corpus QO améliore significativement les résultats pour les systèmes LiaNE et OpenNLP, par contre il est surprenant de constater que le système Standford NER ne tire que peu de bénéfice de cet ajout. Nous pouvons également noter, si nous reprenons les résultats de la table 3, que l'ajout du corpus d'apprentissage OE au corpus d'apprentissage QO augmente dans le cas de LiaNE et Stanford NER la valeur de la précision, du rappel ainsi que de la F mesure par rapport à l'apprentissage sur QO seul. La combinaison des deux corpus a permis d'améliorer la couverture. En conclusion nous pouvons dire que l'ajout de corpus annoté spécifique permet d'améliorer globalement les résultats de tous les outils (F-mesure moyenne : 47,8%) comparativement aux résultats obtenus pour le seul corpus QO (F-mesure moyenne : 43,1%.

Conclusion

Les expériences effectuées dans cette étude sur plusieurs outils de REN et plusieurs corpus d'apprentissage ont permis de mettre en évidence les points suivants : tout d'abord une perte importante de performance est constatée lorsque l'on change le contexte applicatif sur lesquels les modèles ont été appris ; malgré cette baisse de performance, des corpus telles que Quaero sont utiles pour amorcer l'apprentissage de premiers modèles en obtenant des résultats acceptables. Entre le choix du type de document (texte écrit, transcription de l'oral) et la période temporelle visée, les expériences comparatives ont clairement montré l'impact de cette distance temporelle, au regard des plus faibles performances obtenues avec QP plutôt que QO. Enfin, l'ajout d'un corpus d'adaptation, même de taille réduite, permet d'augmenter significativement les performances, même sans autre adaptation au domaine.

Références

Biber D. (1993). Representativeness in corpus design. *Literary and linguistic computing*, **8**(4), 243–257.

Burger J. D., Henderson J. C. & Morgan W. T. (2002). Statistical named entity recognizer adaptation. In *proceedings of the 6th conference on Natural language learning-Volume 20*, p. 1–4 : Association for Computational Linguistics.

Chieu H. L. & Ng H. T. (2002). Named entity recognition : a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, p. 1–7 : ACL.

Favre B., Béchet F. & Noera P. (2005). Robust named entity extraction from large spoken archives. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, p. 491–498, Stroudsburg, PA, USA : Association for Computational Linguistics.

Galliano S., Geoffrois E., Gravier G., Bonastre J.-F., Mostefa D. & Choukri K. (2006). Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *Proceedings of LREC*, volume 6, p. 315–320.

Gildea D. (2001). Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, p. 167–202.

Grishman R. & Sundheim B. (1996). Message understanding conference-6 : A brief history. In *COLING*, volume 96, p. 466–471.

Mansouri A., Suriani L. A. & Mamat A. (2008). Named entity recognition approaches. *International Journal of Computer Science and Network Security*, **8**(2), 339–344.

Marsh E. & Perzanowski D. (1998). Muc-7 evaluation of ie technology : Overview of results.

Nadeau D. & Sekine S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, **30**(1), 3–26.

Nouvel D., Soulet A., Antoine J.-Y., Maurel D. & Friburger N. (2010). Reconnaissance d'entités nommées : enrichissement d'un système à base de connaissances à partir de techniques de fouille de textes. In *Traitement Automatique des Langues Naturelles*.

Poibeau T. (2003). The multilingual named entity recognition framework. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, p. 155–158 : Association for Computational Linguistics.

Rodriquez K. J., Bryant M., Blanke T. & Luszczynska M. (2012). Comparison of named entity recognition tools for raw ocr text.

Rosset S., Grouin C., Fort K., Galibert O., Kahn J. & Zweigenbaum P. (2012). Structured named entities in two distinct press corpora : Contemporary broadcast news and old newspapers. In *Proceedings of the Sixth Linguistic Annotation Workshop*, p. 40–48 : Association for Computational Linguistics.

Rosset S., Grouin C. & Zweigenbaum P. (2011). *Entités nommées structurées : guide d'annotation Quaero*. LIMSI-Centre national de la recherche scientifique.

Sekine S. (1997). The domain dependence of parsing. In *Proceedings of the fifth conference on Applied natural language processing*, p. 96–102 : ACL.

Shaalan K. & Oudah M. (2014). A hybrid approach to arabic named entity recognition. *Journal of Information Science*, **40**(1), 67–87.

Sobhana N., Mitra P. & Ghosh S. (2010). Conditional random field based named entity recognition in geological text. *International Journal of Computer Applications*.

Sun B. (2010). Named entity recognition : Evaluation of existing systems. *Thèse*.

Zweigenbaum P. & Ben Abacha A. (2012). Une étude comparative empirique sur la reconnaissance des entités médicales. *TAL*, **53**.

RENAM: Système de Reconnaissance des Entités Nommées Amazighes

Meryem Talha[1] Siham Boulaknadel[1, 2] Driss Aboutajdine[1]
(1) LRIT, Unité Associée au CNRST (URAC 29), Faculté des Sciences, Mohammed V-Agdal, Rabat, Maroc
(2) IRCAM, Avenue Allal El Fassi, Madinat Al Irfane, Rabat-Instituts, Maroc
meriem.talha@gmail.com, boulaknadel@ircam.ma, aboutaj@fsr.ac.ma

Résumé. La reconnaissance des Entités Nommées (REN) en langue amazighe est un prétraitement potentiellement utile pour de nombreuses applications du traitement de la langue amazighe. Cette tâche représente toutefois un sévère challenge, compte tenu des particularités de cette langue. Dans cet article, nous présentons le premier système d'extraction d'entités nommées amazighes (RENAM) fondé sur une approche symbolique qui utilise le principe de transducteur à états finis disponible sous la plateforme GATE.

Abstract. Named Entity Recognition (NER) for Amazigh language is a potentially useful pretreatment for many processing applications for the Amazigh language. However, this task represents a tough challenge, given the specificities of this language. In this paper, we present (NERAM) the first named entity system for the Amazigh language based on a symbolic approach that uses linguistic rules built manually by using an information extraction tool available within the platform GATE.

Mots-clés : Reconnaissance des entités nommées (REN), Langue Amazighe, Règles d'annotation, JAPE, GATE.
Keywords: Named Entities Recognition (NER), Amazigh Language, Annotation Rules, JAPE, GATE.

1 Introduction

La langue amazighe fait partie des langues chamito-sémitiques ou encore appelé afro-asiatiques (Cohen 2007, Chaker 1989) qu'on soit au Maroc ou ailleurs. La langue Amazighe est maintenant une langue qui possède tous ses attributs: dotée d'une graphie officielle, un codage propre dans le standard Unicode, une grammaire, une orthographe et un vocabulaire très riche.

La tâche REN est une sous tâche du domaine d'extraction d'information consistant à identifier et à catégoriser certaines expressions linguistiques autonomes et mono-référentielles (Ehrmann, 2008). Certaines langues ont éveillé beaucoup d'intérêt, notamment à travers les campagnes d'évaluation telles que CONLL (Tjong Kim Sang, 2002) pour l'espagnol et l'allemand, MUC (Grishman et Sundheim, 1996) pour l'anglais et le japonais, et ESTER (Galliano et al., 2009) pour le français. Toutefois, l'Amazighe étant une langue peu dotée en terme de ressources linguistiques informatisées, les travaux sur de la reconnaissance des entités nommées sont une opportunité pour la valorisation de cette langue dans la société de l'information.

C'est dans cette optique que se situe le travail que nous présentons dont le but est de détecter et extraire dans des textes amazighs les entités nommées pertinentes et ceci en développant le premier système de reconnaissance d'entités nommées (RENAM) qui servira à des applications plus spécifiques. Notre système est fondé sur une approche symbolique où l'extraction s'effectue en se basant sur un ensemble de lexiques de noms et des règles construites manuellement en exploitant l'outil d'extraction des entités nommées disponible sous la plateforme GATE[1].

Notre article est structuré comme suit : la première section présente un état de l'art des approches d'extraction des entités nommées, la deuxième examine les difficultés entravant l'extraction des entités nommées en amazighe, la troisième aborde les particularités de la plateforme choisie ainsi que la méthodologie utilisée, tandis que la discussion

[1] http ://gate.ac.uk

des résultats obtenus est l'objet d'étude de la quatrième section. Finalement, la dernière section est consacrée à la conclusion et perspectives.

2 Approches d'extraction d'entités nommées

La REN constitue un champ de recherche très actif. Différentes approches existent, l'extraction fondée sur des démarches linguistiques ou encore nommées symboliques, approches statistiques ou à base d'apprentissage, avec une apparition d'une approche hybride. La première approche s'appuie sur l'utilisation de grammaires formelles construites à la main. Elle se fonde sur la description des EN grâce à des règles qui exploitent des marqueurs lexicaux, des dictionnaires de noms propres et parfois un étiquetage syntaxique. Les marqueurs lexicaux, il s'agit d'indices qui encadrent, l'entité nommée et qui permettent souvent de dévoiler sa présence. D'autre part, les dictionnaires de noms propres groupent généralement une liste des noms et des prénoms les plus fréquents, des noms de localisations, et parfois des noms d'organisations. Ces dictionnaires sont fréquemment utilisés dans les systèmes à base de règles comme ils peuvent avoir recours aux systèmes à base d'apprentissage. Les ressources mentionnées renforcent l'élaboration des règles et des patrons linguistiques qui spécifient les contextes d'apparition de telle entité. Pour les langues peu dotées, cette approche semble la plus adéquate. Dans le cadre des approches linguistiques, nous citons quelques systèmes de REN arabes à savoir : les travaux de Shaalan et Raza (Shaalan et Raza, 2008) qui ont développé leur système NERA permettant d'extraire dix types d'EN. Ce système repose sur l'utilisation d'un ensemble de dictionnaires d'EN et sur une grammaire sous forme d'expressions régulières pour la reconnaissance des EN. Le meilleur taux F-mesure acquis par ce système est 98.6%. Suivant presque le même principe, les travaux de Zaghouani (Zaghouani et al., 2010) ont présenté un module de repérage des EN à base de règles pour la langue arabe, la seule différence c'est qu'ils ont procédé à une première étape de prétraitement lexicale qui prépare le texte pour son analyse linguistique, ce module a été évalué sur un corpus de presse. Dans le même contexte des langues peu dotées nous évoquons le premier système élaboré pour la langue Iban (Malaisie) par Soo-Fong Yong, Bali Ranaivo-Malançon et Alvin Yeo Wee (Soo-Fong Yong, Bali Ranaivo-Malançon et Alvin Yeo Wee, 2011), il consiste à employer des listes des noms propres et d'un ensemble de règles rédigées manuellement pour permettre l'extraction des entités nommées Iban, ils ont obtenu un F-mesure qui est égal à 76,4%. La seconde démarche fait usage de techniques statistiques ou encore dites à base d'apprentissage pour apprendre des spécifiés sur de larges corpus de textes où les entités-cibles ont été auparavant étiquetées nommés ainsi corpus d'apprentissage, et par la suite adapter un algorithme d'apprentissage qui va permettre d'élaborer automatiquement une base de connaissances à l'aide de plusieurs modèles numériques (CRF, SVM, HMM …). Cette méthode a été envisagée pour avoir une certaine intelligence lors de la prise des décisions, ce sont principalement certains paramètres qui peuvent être manipulés dans le but d'améliorer les résultats du système, ce qui n'est pas le cas pour les approches symboliques qui n'appliquent que les règles préalablement injectées. Benajiba et al (Benajiba et al., 2009) ont conçu une technique d'apprentissage SVM pour leur système de REN en se servant d'un ensemble de particularités de la langue arabe. Ce système a produit une F-mesure de 82.71%. Pour les langues peu doté comme le Telugu P Srikanth et Kavi Narayana Murthy (P Srikanth & Kavi Narayana Murthy 2008) ont utilisé la technique CRF pour extraire les entités nommées et ils ont obtenu un score de f-mesure qui est égal à 92%, pour le Bengali en utilisant le SVM, Asif Ekbal et Sivaji Bandyopadhyay (Asif Ekbal & Sivaji Bandyopadhyay 2008) ont réussi à avoir un résultat dont la valeur de f-mesure est de 91,8%. Les deux approches qu'on a cité auparavant ont fait l'apparition d'une troisième approche qui représente une combinaison de ses antécédents, elle utilise des règles écrites manuellement mais construisent aussi une partie de ses règles en se basant sur des informations syntaxiques et des informations sur le discours extraites de données d'apprentissage grâce à des algorithmes d'apprentissage, des arbres de décisions. Abuleil (Abuleil, 2006) a adopté une approche hybride pour l'extraction des entités en arabes, en tirant profit des approches symboliques et d'apprentissage.

3 Difficultés entravant l'Extraction d'Entités Nommées Amazighes

La langue Amazighe est composée de 27 consonnes, 2 semi-consonnes, 3 voyelles pleines et une voyelle neutre. Elle présente une morphologie riche et complexe. Les mots peuvent être classés en trois catégories morphosyntaxiques: Nom, Verbe et Particules[2] (Boukhris et al., 2008). La structure basique de la phrase simple est : Sujet – verbe – complément. Cependant, les travaux liés à la REN de la langue amazighe sont encore à l'état naissant en raison de:

[2] Elles sont un ensemble de mots Amazighs qui ne sont ni des noms, ni des verbes, et jouent un rôle d'indicateurs grammaticaux au sein d'une phrase. Cet ensemble est constitué de plusieurs éléments à savoir: les particules d'aspect, d'orientation et de négation; les pronoms; les adverbes; les prépositions; les subordonnants et les conjonctions.

– L'absence de la distinction majuscule/minuscule : c'est un obstacle majeur pour la langue amazighe. En fait, la REN pour certaines langues comme les langues indo-européennes se base principalement sur la présence des lettres majuscules qui est un indicateur très utile pour identifier les noms propres dans les langues utilisant l'alphabet latin. Il n'y a pas de majuscule, ni au début ni à l'initiale des noms propres amazighe.
– L'amazighe se caractérise par le manque de ressources dictionnairiques, de répertoires toponymiques, de ressources langagières et outils du TAL, à savoir les étiqueteurs, les analyseurs morphologiques.
– Il est un fait que la langue amazighe est très agglutinative ayant une morphologie dérivationnelle et flexionnelle assez complexe.
– Similairement à d'autres langues naturelles, l'amazighe présente des incertitudes au niveau des classes grammaticales. En effet, la même forme convient à nombreuses catégories grammaticales, cela dépend du contexte dans la phrase. Par exemple, ⵉⵍⵍⵉ [illi] peut être considéré comme verbe à l'accompli positif, il signifie «il existe», ou comme nom de parenté «ma fille».
Les noms propres au niveau de la langue amazighe sont extrêmement nombreux, ont de nombreuses variantes ainsi ils sont difficiles à détecter sans la présence d'un lexique.

4 Système de Reconnaissance d'Entités Nommées Amazighes (RENAM)

La tâche de REN amazighes apparaît fondamentale pour diverses applications participant à l'analyse du contenu des textes amazighes. Dans cette contribution, nous nous intéressons à développer un système d'analyse de textes amazighes permettant le repérage et la catégorisation des entités nommées en fonction de types sémantiques prédéfinis. Pour respecter les formats d'annotation standard, nous avons adopté le standard d'étiquetage proposé par la conférence MUC (Message Understanding *Conference*) pour les entités nommées. Par conséquent les EN appartiennent à trois classes majeures à savoir la catégorie **ENAMEX** qui comprend les types « personne », « organisation », et « localisation », la catégorie **TIMEX** qui inclut les types « date », « heure », et finalement la catégorie **NUMEX** qui comprend « montants financiers », « pourcentages » tout en exploitant la plate-forme GATE.

4.1 Plateforme GATE

La plateforme d'ingénierie textuelle **GATE** (General Architecture for Text Engineering) (Cunningham et al. 2002) est une infrastructure de développement de traitement du langage humain développée par l'Université de Sheffield et est exploitée dans une vaste variété de recherche et de projets de développement incluant l'extraction de connaissances pour l'anglais, l'espagnol, le chinois, l'arabe, le français, l'allemand, l'hindi, le cebuano, le roumain, le russe. Nous avons choisi cet environnement car il permet d'implémenter l'architecture souhaitée et il peut être utilisé pour l'exploitation des traitements linguistiques dans diverses applications. Le module d'extraction proposé dans GATE est réalisé selon une approche symbolique basé sur le formalisme **JAPE** (Java Annotation Patterns Engine) permettant de définir les contextes d'apparition des unités à extraire dans le but de les repérer et les annoter.

4.2 Architecture du système

Etant donné la non-disponibilité d'un large corpus pour la langue amazighe, nous avons créé le notre. Notre système d'extraction d'entités nommées est un système de détection et de typage des entités d'intérêt dans un texte. Notre but se limite à l'extraction des entités de type «Personne», «Organisation», et «Localisation», dans des textes écrits en amazighe, en utilisant une approche symbolique en créant manuellement des règles pour extraire les informations qui nous intéressent.

Notre système RENAM se compose de deux grandes phases successives (Figure 1): la première consiste à la préparation des données et la deuxième consiste à la reconnaissance des entités nommées et classification de ces dernières dans des catégories sémantiques convenables (personne, localisation, organisation).

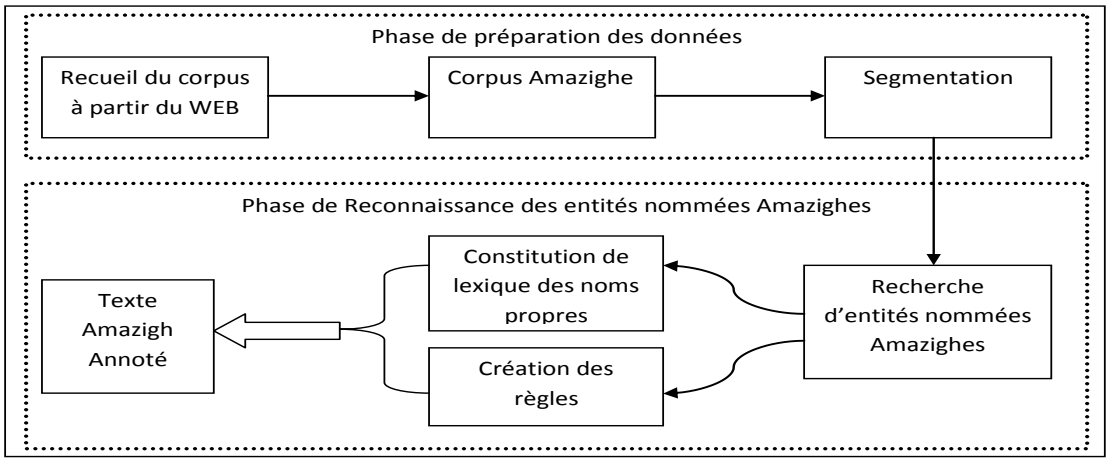


FIGURE 1: Architecture du Système d'Extraction des Entités Nommées Amazighes

– Phase de préparation des données

1. Recueil du corpus

La construction d'un système d'extraction d'entités nommées exige, d'abord, de rassembler un nombre suffisant de textes qui serviront non seulement de corpus d'observation (pour constituer les règles) mais également de corpus de test. Le contexte de cette contribution nous a naturellement guidé vers la collecte de textes amazighes journalistiques à partir du site web « mapamazighe[3]». Le corpus contient toute l'actualité sur les activités royales de SM le Roi Mohammed VI, on dispose actuellement de 200 articles écrits en amazighe, que nous avons convertis du format « html » au format « txt », cumulant un total de 50997 mots, nous signalons aussi qu'outre ce corpus inclut 4866 chiffres.

2. Segmentation du corpus

Dans cette phase, nous séparons le texte amazigh en des phrases, nous avons en total 1214 phrases, et ensuite nous séparons les phrases en des mots. Dans le texte, les phrases sont fragmentées selon la présence des ponctuations et le retour à la ligne, comme dans le cas dans les langues française et anglaise. Ces séparateurs définissent les modèles de caractères marquant la fin des phrases.

– Phase d'extraction des entités nommées

3. Constitution de lexique des noms propres

Dans le but de pouvoir reconnaître automatiquement les entités nommées de type personne, organisations et localisation, nous avons commencé par l'élaboration manuelle d'un lexique provenant de plusieurs sites Web.
Pour les entités nommées de type « personne», nous avons pu construire une liste d'environ 1650 entrées de noms amazighes et noms transcrits en amazighe[4]. Pour les entités nommées de type « localisation », nous nous sommes inspirés de la classification faite par (Piton et Maurel, 2004) qui considère comme type « localisation » ou toponyme : les pays, villes, fleuves, montagnes, océans et mers. Ainsi, nous avons élaboré un lexique à partir de wikipedia (nous avons procédé à la transliteration des entités françaises extraites à partir de wikipédia vers l'amazighe), et le site officiel de l'IRCAM qui contient 1970 entités de type « localisation ». A l'instar de la procédure d'élaboration de lexiques des deux entités nommées précédentes, l'identification des entités nommées de type organisation a commencé par le développement d'un lexique qui contient 120 noms d'organisation, à partir de notre corpus qui contient un nombre assez important des noms d'organisations, et les sites web officiels des organisations en question.
Nous avons aussi élaboré un lexique des marqueurs lexicaux de type localisation avec 55 entrées, de type personne avec 84 entrées et de type organisation avec 70 entrées.

[3] http://www.mapamazighe.ma/am/

[4] www.dsic.upv.es/%7Eybenajiba/resources/ANERGazet.zip

4. Développement des règles linguistiques

Pour la construction de nos règles linguistiques pour le traitement de l'amazighe, nous nous sommes inspirés des travaux de (McDonald, 1996) dans le domaine qui consiste à déterminer les indices d'apparition des entités à extraire. Ainsi, ces derniers font partie intégrante de l'entité nommée comme le prénom pour une entité de type «Personne » ou correspondent au contexte textuel de l'entité pouvant indiquer son type. La découverte des marqueurs lexicaux s'est faite par observation du corpus décrit précédemment.

Règles de type « Personne » :

Les règles pour la détermination de l'entité personne fait usage à la présence :

- Des civilités à titre d'exemple : ⵎⴰⵙⵙ [mass] « Monsieur », ⵜⴰⵎⵎⴰ [mast] « Mme », ⵜⴰⵎⴰⵙⵙⴰ [massa] « Mme », ⵍⴰⵍⵍⴰ [lalla] « Mme », ⵎⵓⵍⴰⵢ [moulay] « My » …
- Les titres de toute sortes tels que :
  - Titres politiques (40 entrées) : ⴰⵎⴰⵡⴰⵙ [amawas] « ministre »
  - Titre honorifiques (14 entrées) : ⴱⴰⴱ ⵏ ⵡⴰⵏⵏⵓⵔ [bab n waddur] « Sa Majesté »
  - Titre religieux (8 entrées) : ⵛⵉⵅ [chikh] « Cheikh »
- Les métiers ou professions (… entrées) : ⵍⴽⵓⵍⵓⵏⵉⵍ [lkulunil] « le colonel »

Pour définir le lien de parenté, chaque culture a son propre système de nommage, par exemple ⵎⵓⵃⵎⵎⴷ ⴱⵏ ⵣⴰⵢⴷ [muhmmd bn zayd] « Mohamed Ibn Zayd », le modèle suivant indique que cette séquence est un nom de personne :

<NomPropre> <ⴱⵏ (bn, fils) > <NomPropre> ➔ Nom= « entité personne ».

Règles de type « Localisation » :

Les règles linguistiques pour la reconnaissance des entités de *type « localisation »* sont écrites en se basant sur des prépositions (15 entrées) qui accompagnent toujours les noms de lieux comme par exemple ⴷⴰⵔ [dar] « en direction de », ⵖⵔ [ghr] « vers », ⵊⴰⵔ…ⴷ… [jar…d…] « entre… et… »…

En outre, nous avons construit une liste de marqueurs lexicaux comme par exemple, ⵜⵎⴷⵉⵏⵜ [tmdint] « ville ». Ces deniers sont exploités pour la description des règles de reconnaissance des entités de *type « localisation ».*

Par exemple, ⴰⵙⵉⴼ ⵏ ⵙⴱⵓ [Asif n sbu] « rivière de sbou », ce qui signifie que tout mot qui vient après un préfixe de localisation (dans ce cas ⴰⵙⵉⴼ [asif] « rivière ») sera annoté comme étant un nom de lieu.

Règles de type « Organisation » :

Nous avons élaboré une liste des marqueurs lexicaux comme par exemple ⵜⴰⵎⴰⵡⴰⵙⵜ [amawast] « Ministère » pour la description des règles de reconnaissance des entités de *type « organisation »*

Tableau récapitulatif de règles construites :

Entités nommées	Exemple	Nombre de règles
Personne	ⵎⴰⵙⵙ(Monsieur), ⵜⴰⵎⵎⴰ(Madame), ⵙⴰⵙ ⵏ ⵡⴰⵏⵏⵓⵔ (Sa majesté)…	10
Organisation	ⵜⵉⵎⵙⵙⵏ(Ecole), ⵜⴰⵎⴰⵡⴰⵙⵜ(Ministère) …	4
Localisation	ⴰⵙⵉⴼ(Fleuve), ⵜⵎⴷⵉⵏⵜ(Ville) …	5

TABLE 1 : Tableau récapitulatif

5 Evaluation des résultats

Une dernière étape de cette contribution est l'évaluation des résultats de notre outil d'extraction des entités nommées amazighes. Ainsi, nous présentons, dans cette section, les différentes étapes de notre évaluation et les conclusions que nous avons pu faire en vu des résultats obtenus.

5.1 Protocole expérimental

L'évaluation de notre système a, ainsi, permis d'établir plusieurs choix visant le type d'évaluation à mettre en place. Deux solutions se sont alors présentées : exploiter les données d'une campagne d'évaluation existante ou créer notre propre système d'évaluation. Le premier cas n'étant pas possible puisqu'il n'existe pas un corpus de la langue amazighe, où les entités nommées ont été préalablement annotées. En conséquence, nous avons envisagé la deuxième solution, qui est le développement de notre système d'évaluation.

5.2 Analyse des résultats

Dans notre évaluation, nous avons choisi les métriques qui nous permettront d'évaluer les résultats de nos travaux et ainsi mesurer les performances du système proposé.

La précision (mesure de qualité), le rappel (mesure de quantité) et la F-mesure (synthèse de Rappel et de précision) (*Van Rijsbergen, 1979*), ont été choisis pour leur exploitation fréquente dans le domaine du TAL.

$$\text{Rappel} = \frac{nombre\ d'entités\ correctement\ étiquetées}{nombre\ d'entités}$$

$$\text{Précision} = \frac{nombre\ d'entités\ correctement\ étiquetées}{nombre\ d'entités\ étiquetées}$$

$$\text{F-Mesure} = \frac{2 \times Précision \times Rappel}{Précision + Rappel}$$

Nous avons effectué une évaluation sur notre corpus « Activités Royales » que nous avons déjà décrit, et cela a donné les résultats suivants :

Entités nommées	Précision	Rappel	F-mesure
Personne	64%	63%	64%
Organisation	27%	71%	40%
Localisation	82%	81%	82%

TABLE 1 : Evaluation de notre système de reconnaissance des entités nommées

D'après la Table 2, les résultats que nous avons obtenus sont plus ou moins satisfaisants et encourageants. Cependant notre outil fonctionne légèrement moins bien pour l'entité nommée « organisation ».

Les principales causes de cette déficience sont :

- On a remarqué que les noms d'organisation sont très peu utilisés par rapport aux autres entités nommées.

- Les entités nommées complexes ne sont pas correctement extraites. A titre d'exemple ; le mot déclencheur ⵜⴰⵎⴰⵡⴰⵙⵜ [tamawast] « Ministère » qui indique que la séquence qui va suivre représente une entité nommée de type organisation, néanmoins l'absence des informations morphologiques nécessaires rend la tâche de délimitation de l'entité en question plus difficile. Par exemple : ⵜⴰⵎⴰⵡⴰⵙⵜ ⵏ ⵙⴳⵎⵉ ⴰⵏⴰⵎⵓⵔ [tamawast n usgmi anamur] « Ministère de l'éducation nationale ». Notre outil ne va reconnaitre que l'entité « ⵜⴰⵎⴰⵡⴰⵙⵜ ⵏ ⵙⴳⵎⵉ », parce qu'il n y a pas de critère d'arrêt selon chaque entité.

– La prise en compte des variantes orthographiques des noms propres transcrits en l'absence de conventions pour leurs écritures (notamment pour les noms de lieux). En amazighe, la translittération et la transcription des noms propres étrangers n'obéissent pas à des règles d'écritures par exemple ⴰⴱⵓ ⴷⴰⴱⵉ [abu Dabi] « Abu Dhabi » ou ⴰⴱⵓⴷⴰⴱⵉ [abudabi] « Abudhabi »).

6 Conclusion et perspectives

L'objectif principal de cette contribution est de reconnaître les entités nommées de types (personne, localisations, organisations,) dans les textes amazighes. De ce fait, nous avons élaboré dans un premier temps, un système de reconnaissance d'entités nommées pour l'amazighe, fondé sur une approche symbolique qui se base sur un ensemble de règles linguistiques construites manuellement et sur l'élaboration des lexiques de noms en se servant de la plateforme GATE. La valeur de F-mesure obtenue par notre méthode est assez encourageante malgré la présence des problèmes qu'on a décrit auparavant. Dans un futur immédiat, nous envisageons améliorer la qualité des règles d'extraction d'entités nommées que nous avons établies, étendre et enrichir la structure de nos lexiques de noms afin de couvrir le maximum des entités nommées et pour aboutir à un taux de reconnaissance plus élevé, étendre la taille de notre corpus, ensuite passer au traitement des deux autres catégories TIMEX (Date…) et NUMEX (pourcentage, …).

Références

BENAJIBA Y., ROSSO P. (2008). Arabic Named Entity Recognition using Conditional Random Fields. *In Proceedings of Workshop on HLT and NLP within the Arabic World*, *LREC*.

BICKEL M., MILLER S., SCHWARTZ R., WEISCHEDEL R. (1997). "Nymble: a high-performance learning name-finder". *In Proceedings of the ANLP 97.*

BORTHWICK A., STERLING J., AGICHTEIN E., GRISHMAN R. (1998). "Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition". *In Proceedings of the WVLC 98.*

BOUKHRIS F., BOUMALK A., ELMOUJAHID E., SOUIFI H. (2008). La nouvelle grammaire de l'amazighe. Rabat, Maroc: IRCAM

CHAKER S. (1984). Textes en linguistique berbère - introduction au domaine berbère, *éditions du CNRS*, 232-242.

COHEN D. (2007). Chamito-sémitiques (langues). *In Encyclopædia Universalis*.

CUNNINGHAM H. (1999). Information Extraction: a User Guide (revised version), *Research Memorandum*, Department of Computer Science, University of Sheffield

CUNNINGHAM H., MAYNARD D., BONTCHEVA K., TABLAN V., URSU C. (2002). The GATE User Guide.

EHRMANN M. (2008). Les entités nommées, de la linguistique au TAL : statut théorique et méthodes de désambiguïsation. *PhD thesis*, Université Paris 7

FONG Y., BALI R., WEE A. (2011). NERSIL - the Named-Entity Recognition System for Iban Language. *PACLIC 2011:* 549-558

GAHBICHE B S., BONNEAU H., MAYNARD D., LAVERGNE T, YVON F. (2012), Repérage des entités nommées pour l'arabe, *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2: TALN, 487–494*

MAYNARD D. (*2011*). Developing Language Processing Components with GATE, Version 6, *http://gate.ac.uk/sale/tao/tao.pdf.*

MCDONALD D. (1996). Internal and external evidence in the identification and semantic categorization of proper names. *In B. Boguraev and J. Pustejovsky, editors, Corpus Processing for Lexical Acquisition,* 21–39.

OANA H., BONTCHEVA K., MAYNARD D., TABLAN V., CUNNINGHAM H. (2003). Named entity Recognition in Romanian using Gate. *RANLP 2003 Workshop on information Extraction for slavonic and other central and eastern European languages*, Borovets : Bulgaria

GRISHMAN R. (1995). The NYU system for MUC-6 or Where's the Syntax. *In the proceedings of Sixth Message Understanding Conference (MUC-6),* 167-195

SHAALAN K., RAZA H. (2009). NERA : Named entity recognition for arabic. *Journal OF the American Society for Information Science and Technology*, 1652–1663.

TAKEUCHI K., COLLIER N. (2002). Use of support vector machines in extended named entity. *In*: *Proc*. *CoNLL-2002*

TJONG K. S., DEMEULDER F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, *in: Proceedings of the 7th Conference on Natural Language Learning*, Edmonton, Canada, 142–147.

WAKAO T., GAIZAUSKAS R., WILKS Y. (1996). Evaluation of an algorithm for the recognition and classification of proper names. *In Proceedings of COLING-96*.

ZAGHOUANI W., POULIQUEN B., EBRAHIM M., STEINBERGER R. (2010). Adapting a resource-light highly multilingual named entity recognition system to arabic. *Proceedings of the Seventh conference on International Language Resources and Evaluation* (*LREC'10*), 563–567.

De la quenelle culinaire à la quenelle politique : identification de changements sémantiques à l'aide des Topic Models

Ingrid Falk Delphine Bernhard Christophe Gérard
LiLPa, Université de Strasbourg
ifalk, dbernhard, christophegerard@unistra.fr

Résumé. Dans cet article nous employons le « topic modeling » pour explorer des chemins vers la détection automatique de l'apparition de nouveaux sens pour des mots connus. Nous appliquons les méthodes introduites dans (Lau *et al.*, 2012, 2014) à un cas de néologie sémantique récent, l'apparition du nouveau sens de *geste* pour le mot « quenelle ». Nos expériences mettent en évidence le potentiel de cette approche pour l'apprentissage des sens du mot, l'alignement des topics à des sens de dictionnaire et enfin la détection de nouveaux sens.

Abstract. In this study we explore topic modeling for the automatic detection of new senses of known words. We apply methods developed in previous work for English (Lau *et al.*, 2012, 2014) on a recent case of new word sense induction in French, namely the appearence of the new meaning of *gesture* for the word « quenelle ». Our experiments illustrate the potential of this approach at learning word senses, aligning the topics with dictionary senses and finally at detecting the new senses.

Mots-clés : topic models, induction de sens, néologie sémantique.

Keywords: topic models, word sense induction, semantic neologism.

1 Introduction

Dans cet article nous nous proposons de contribuer à la détection automatique de la néologie sémantique qui, traditionnellement, est définie comme un changement de sens sans changement de forme. Dans ce but, nous appliquons les méthodes basées sur les Topic Models, présentées dans (Lau *et al.*, 2012, 2014), à l'identification automatique du nouveau sens du signifiant « quenelle ». Ces méthodes sont très récentes et ont l'avantage d'être relativement simples à mettre en œuvre en comparaison à la majorité des travaux proposés jusqu'alors pour la détection automatique de la néologie sémantique. Toutefois, elles n'ont pour l'heure pas été appliquées au français.

Le signifiant « quenelle », désignant originellement une préparation culinaire, a acquis récemment un sens politique qui a suscité de nombreux écrits sur la Toile. Nous disposons ainsi d'un matériau textuel comprenant d'une part les usages classiques de ce mot (domaine culinaire) et d'autre part de ses usages très récents (domaine politique).

Cette étude ne porte donc que sur un type particulier de néologie sémantique bien distinct d'autres types existants. Le cas du mot « quenelle » ne relève ainsi ni de l'extension de sens (*arriver*), ni de la restriction de sens (*pondre*), ni de la conversion (*centenaire*, adj > *centenaire*, nom), ni de l'antonomase (un *bordeaux*). Bien qu'il soit possible d'argumenter dans le sens d'une création homonymique, il semble que la symbolique liée au geste raciste de la quenelle motive d'en parler en termes de création métaphorique, par exemple sur le modèle de « caviar » (domaine culinaire > domaine sportif).

2 Travaux liés

L'identification de la néologie sémantique reste problématique, et les méthodes proposées sont donc essentiellement semi-automatiques ou limitées à l'analyse manuelle d'exemples précis à l'aide de mesures statistiques. La méthode proposée par Reutenauer (2012) procède par annotation du corpus en sémèmes (lemmes de mots pleins issus des définitions lexicographiques fournies par le Trésor de la Langue Française Informatisé pour chaque mot plein du corpus). Les cooccurrents

les plus significatifs d'un mot étudié sont ensuite extraits par calcul de la spécificité (Reutenauer *et al.*, 2010). Si, en l'occurrence, le couplage de la sémantique différentielle et de la linguistique informatique permet bien de rendre compte d'une évolution de sens (extension / restriction), la sélection du candidat à la néologie reste manuelle. Les travaux de Boussidan & Ploux (2011) reposent sur le modèle ACOM (Ji *et al.*, 2003) qui utilise l'analyse factorielle pour fournir des représentations géométriques de la cooccurrence des mots dans un corpus. L'étude des patrons de cooccurrence permet de détecter des changements de sens. L'analyse des mots cooccurrents pour la détection de la néologie sémantique est également proposée par Cabré & Nazar (2011).

Dans notre cas précis le changement de domaine du mot « quenelle » se traduit par un changement de thème. Les méthodes très récentes basées sur les Topic Models permettent de modéliser de manière extrêmement simple le contexte thématique et son évolution. Elles paraissent ainsi particulièrement adaptées à notre cas d'étude, même si pour l'heure elles n'ont à notre connaissance été appliquées qu'à l'anglais.

Un « topic model » est un modèle probabiliste permettant de déterminer des sujets ou thèmes abstraits dans une collection de documents. Dans un tel modèle un document est considéré comme composé d'un ou plusieurs « topics » qui à leur tour sont vus comme distributions de probabilité sur l'espace des mots. Un « topic model » est génératif dans la mesure où il spécifie une procédure probabiliste pour la génération de documents. Pour créer un nouveau document on choisit d'abord une distribution de topics. Ensuite on choisit aléatoirement pour chaque mot de ce document un topic suivant cette distribution, et finalement un mot du topic sélectionné. Des techniques statistiques standard permettent d'inverser ce processus et d'inférer l'ensemble des topics ayant mené, selon ce paradigme, à la collection de documents observée (Steyvers & Griffiths, 2007).

Le modèle utilisé le plus fréquemment en TAL est LDA – Latent Dirichlet Allocation (Blei *et al.*, 2003), qui présente l'inconvénient que l'utilisateur doit fixer le nombre de topics à priori. Pour les expériences que nous présentons ici nous appliquons un modèle non-paramétré appelé HDP – Hierarchical Dirichlet Process (Teh *et al.*, 2006), où le nombre de topics est inféré en même temps que les topics.

Lau *et al.* (2012, 2014) appliquent le HDP à une tâche de détection des sens d'un mot. Leur approche peut se résumer comme suit. Dans un premier temps on constitue pour le mot observé un corpus composé de phrases contenant ce mot, phrases qui représentent les usages du mot étudié. On applique ensuite le topic modeling et on compare les topics inférés aux sens attestés dans des dictionnaires. Dans un deuxième temps ce corpus est divisé en un corpus de référence, représentant les usages connus, et un corpus dit nouveau supposé de contenir les usages avec un nouveau sens. On est ainsi en mesure d'étudier et comparer les topics (sens) avec lesquels le mot est employé dans les deux corpus.

3 Ressources

Corpus. Le corpus à la base de nos expériences se constitue d'une part des 164 paragraphes du corpus Le Monde (1987-2006) contenant le mot « quenelle ». Cette partie du corpus (*REF* pour la suite) représente l'usage classique, dans le domaine culinaire, du mot « quenelle ». D'autre part nous avons extrait 342 paragraphes contenant le mot « quenelle » des journaux suivants, disponibles en ligne : La Croix, Le Figaro, Le Monde, L'Équipe, Les Echos et Libération. Ces paragraphes proviennent de 160 articles datés entre septembre 2013 et janvier 2014.

Prétraitement. Pour l'apprentissage des Topic Models nous ne prenons en compte que les noms communs, verbes, adjectifs et adverbes présents dans Morphalou (Romary *et al.*, 2004). Si dans Morphalou une forme n'a qu'un seul lemme, elle est remplacée par celui-ci. La taille du vocabulaire obtenu de cette manière est de 2 430 mots et la taille moyenne des paragraphes collectés passe de 77.35 mots avant prétraitement à 25.91 mots après.

Dictionnaires. Dans la plupart des dictionnaires en ligne (le TLFi, Larousse et Reverso) seul le sens culinaire de « quenelle » est attesté. Voici par exemple la définition du TLFi :

> Préparation, en forme de boulette ou de petit cylindre, à base de viande, d'abats, ou de poisson finement hachés, incorporée à une pâte de farine ou de mie de pain, que l'on sert telle quelle accompagnée d'une sauce ou dans une garniture de pâté chaud : vol-au-vent, bouchée à la reine.

Seuls Wiktionary et Wikipedia attestent aussi le nouveau sens de *geste*. Les articles de Wikipédia nous semblant plus exhaustifs, c'est ceux-ci que nous utilisons dans nos expériences.

Hypothèses de travail. Puisque le sens *geste* n'est apparu qu'en 2005 [1] nous considérons que le corpus *REF* ne contient pas d'usages de « quenelle » dans le sens *geste*. Par contre, le corpus *NOUV* peut contenir à priori des usages dans les deux sens, *culinaire* et *geste*. Cependant, les fréquences d'usage (164 usages sur 10 ans pour le corpus *REF* contre 342 usages sur quelques mois pour le corpus *NOUV*) laissent supposer une prépondérance de l'usage *geste*.

4 Expériences et discussion des résultats

Du corpus regroupant *REF* et *NOUV* et en appliquant un « Hierarchichal Dirichlet Process » (HDP) nous avons inféré un « topic model » [2]. Rappelons (Section 2) que ce type de modèle apprend le nombre de topics en même temps que la distribution des mots en topics.

Les 6 topics résultants sont présentés dans le Tableau 1. Ces extraits sont déjà suffisants pour constater que les topics 3 et 5 sont clairement associés au sens *culinaire* de « quenelle » alors que les topics 1, 2 et 4 relèvent plutôt du sens *geste*. Par contre le topic 6 ne présente pas une interprétation aussi évidente.

T1	quenelle geste antisémite salut photo nazi bras humoriste français effectuer ...
T2	quenelle jean marier marque fion fond glisser petite sionisme déposer ...
T3	quenelle carte menu plat brochet cuisine rue compter veau dessert ...
T4	quenelle spectacle public geste jeune interdiction antisémite humoriste monde ordre ...
T5	quenelle brochet pain épices grand sauce chair cuisine écrevisse beurre ...
T6	vide quenelle couverts charges cru debout derrière docteur effet émission ...

TABLE 1: Les 6 topics appris sur le corpus *REF+NOUV*. Nous affichons les 10 mots les plus probables de la distribution.

En s'appuyant sur ces topics, le topic modeling permet d'analyser les textes du corpus et de déterminer la présence et la proportion des topics présents dans chaque paragraphe. Un exemple est présenté en Tableau 2.

Paragraphe Du côté associatif, la Ligue des droits de l'Homme (LDH) a accueilli la circulaire avec prudence, redoutant les risques d'une interdiction préalable qui pourrait « fédérer autour de Dieudonné une sympathie réactionnelle de ceux qui se considèrent opprimés ». En revanche, la LDH encourage à poursuivre en justice chaque atteinte à la loi de Dieudonné, déjà condamné à de multiples reprises pour antisémitisme ou injure raciale. SOS Racisme et l'Union des étudiants juifs de France (UEJF) ont emboîté le pas, annonçant qu'ils poursuivraient désormais toutes les "quenelles" effectuées dans des lieux où elles "ne laissent pas de doute" sur leur caractère antisémite. Le Conseil représentatif des institutions juives de France (Crif) a quant à lui appelé à la « mobilisation » dans la vingtaine de villes où l'humoriste doit se produire.
Après prétraitement : côté associatif ligue droit homme accueillir circulaire redouter risques interdiction fédérer autour sympathie considérer revanche encourager poursuivre justice atteinte loi condamné multiple reprises antisémitisme injure racial racisme union étudiant juif annoncer poursuivre quenelle effectuer lieu laisser doute caractère antisémite conseil représentatif institution juives appelé mobilisation ville humoriste produire
Distribution des topics : T1 56.00%, T2 44.00%

TABLE 2: Exemple de distribution de topics dans un paragraphe. Interprétation : La génération de ce texte s'explique selon le modèle par un tirage de mots du topic T1 et du topic T2 avec des pourcentages respectifs de 56% et 44%.

Dans ce qui suit nous allons d'abord analyser si ces topics correspondent aux usages de « quenelle » dans notre corpus (Section 4.1) avant d'explorer d'autres méthodes et mesures introduites dans (Lau *et al.*, 2012, 2014), plus spécifiquement : l'alignement sens ↔ topic (Section 4.2), la détermination du sens prédominant (Section 4.3) et enfin la détection de sens nouveaux (Section 4.4).

1. Wikipédia, http ://fr.wikipedia.org/wiki/Dieudonné#Le_geste_de_la_.C2.AB_quenelle_.C2.BB_et_autres_signes_de_ralliement, 30/04/2014
2. Pour toutes nos expériences nous nous sommes fortement appuyés sur les logiciels disponibles librement à `https://github.com/jhlau/predom_sense` et `https://github.com/jhlau/hdp-wsi`

4.1 Correspondances topics ↔ usages dans corpus

Le tableau 3 montre pour chaque topic, le nombre de paragraphes pour lesquels ce topic est prédominant. Est considéré prédominant dans un paragraphe le topic avec la plus grande probabilité dans la distribution de topics pour ce paragraphe (ainsi dans le tableau 2 le topic prédominant est le topic T1).

Corpus	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
REF	0	6	**118**	0	**36**	**4**
NOUV	**229**	**32**	9	**42**	29	1
Total	229	38	127	42	65	5

TABLE 3: Topics prédominants par paragraphes.

Nous voyons que la plupart (154 sur 164) des paragraphes du corpus *REF* ont comme topic prédominant les topics T3 et T5, ce qui correspond bien au sens *culinaire* de « quenelle ». D'autre part les topics prédominants de 303 des 342 paragraphes du corpus *NOUV* sont les topics T1, T2 et T4 qui relèvent du sens *geste*.

Les erreurs potentielles d'analyse thématique sont les paragraphes du corpus *NOUV* à topic prédominant *culinaire* et ceux du corpus *REF* à topic prédominant *geste*. Le corpus *REF* n'ayant pas de paragraphes à topic prédominant *geste*, nous nous tournons vers les 38 paragraphes à topic *culinaire* dans le corpus *NOUV*. 7 des 9 paragraphes à topic prédominant T3 suggèrent effectivement un sens *culinaire*. Les 2 paragraphes restants sont trop courts (2 respectivement 3 mots) pour permettre une évaluation du sens [3].

Les 5 des 29 paragraphes à sens prédominant T5, ne correspondant pas à un sens *culinaire* sont :

quenelle continue inquiéter
doublé quenelle
quenelle laisser sale goût
quenelle laisser sale goût
mauvaise quenelle

Si la brièveté de ces textes ou le contexte avec des connotations (ambiguës) *culinaires* (*goût*, *mauvaise* [4]) pourrait éventuellement justifier l'analyse erronée, il faut noter que ces cas sont néanmoins faciles à départager pour un juge humain. Notons aussi que seulement 5 paragraphes ont comme topic prédominant le topic 6, qui était particulièrement difficile à interpréter. Pour conclure nous constatons que globalement l'analyse par les topic models permet de bien départager les deux sens de « quenelle » dans notre corpus.

4.2 Correspondances sens ↔ topic

Dans cette section nous appliquons les méthodes décrites dans (Lau *et al.*, 2014) pour déterminer le sens prédominant d'une collection de documents. Dans ces expériences le sens est représenté par les entrées d'un dictionnaire, en l'occurrence les gloses de Wikipédia. Ces gloses sont converties en distributions (par l'estimation du maximum de vraisemblance) et mises en relation avec les topics. Nous calculons la similarité d'un sens de dictionnaire s_i et d'un topic t_j suivant l'équation suivante (Lau *et al.*, 2014) :

$$sim(s_i, t_j) = 1 - JS(S\|T) = 1 - \frac{1}{2}KL(S\|M) - \frac{1}{2}KL(T\|M) \quad (1)$$

ou S et T sont respectivement les distributions multinomiales de mots pour les sens s_i et topics t_j respectivement ; $M = \frac{1}{2}(S+T)$; et $JS(X\|Y)$ et $KL(X\|Y)$ sont les divergences Jensen-Shannon et Kulbach-Leibler pour les distributions X et Y respectivement.

Nous présentons dans le Tableau 4 la similarité des topics, suivant l'équation (1), aux sens *culinaire* et *geste* de « quenelle » (tel qu'attesté dans Wikipédia). On voit à nouveau que cet alignement automatique correspond très bien à nos observations en Section 4.1. Il se pose cependant encore le problème de détecter automatiquement à partir de quel taux de similarité

3. Il s'agit ici de titres d'articles qui sont déjà courts à l'origine et ont été raccourcis davantage lors du prétraitement.
4. Pour les trois derniers textes il s'agit en fait de jeux de mots.

culinaire	T5	T3	T1	T2	T6	T4	moyenne
	0.266	0.2228	0.1162	0.1141	0.0714	0.0709	0.1436
geste	T1	T2	T4	T5	T3	T6	moyenne
	0.3917	0.3181	0.2559	0.1722	0.1289	0.071	0.223

TABLE 4: Topics les plus similaires aux sens *culinaire* et *geste*, tel qu'attesté dans Wikipédia et suivant équation (1)

un topic ne devrait plus être associé à un sens. Une possibilité serait éventuellement de prendre les valeurs au dessus de la moyenne.

4.3 Sens prédominant

Sur la base de cette similarité et de la distribution des topics dans les paragraphes nous déduisons une distribution des sens (suivant toujours (Lau *et al.*, 2014)) :

$$prevalence(s_i) = \sum_{j=1}^{T} \left(sim(s_i, t_j) \times \frac{f(t_j)}{\sum_{k=1}^{T} f(t_k)} \right) \quad (2)$$

ou T est le nombre de topics et $f(t_j)$ le nombre d'usages (paragraphes) à topic prédominant t_j. Suivant cette formule nous obtenons la prévalence des sens dans le corpus. Nous avons aussi calculé la proportion de paragraphes pour chaque sens selon notre analyse manuelle des données [5] :

sens	**prévalence**	**proportion paragraphes**
culinaire	0.3625	195/506 = 0.3854
geste	0.6375	≈304/506 = 0.6008

Le modèle permet donc d'estimer assez bien la proportion d'usages dans le corpus correspondant à un certain sens (de dictionnaire).

4.4 Détection de topics

Dans cette section nous appliquons une mesure de nouveauté introduite par Lau *et al.* (2012). Ce score permet d'estimer si certains topics sont inédits dans le nouveau corpus. Pour un topic t_i le score de nouveauté Nouv(t_i) (par rapport au mot « quenelle ») est définit par l'équation (3).

$$\text{Nouv}(t_i) = \frac{p_{NOUV}(t_i) - p_{REF}(t_i)}{p_{REF}(t_i)} \quad (3)$$

où $p_{NOUV}(t_i)$ et $p_{REF}(t_i)$ représentent les probabilités (calculées par estimation du maximum de vraisemblance) des topics t_i dans le corpus nouveau et de référence respectivement. Le score de nouveauté est élevé si un topic est beaucoup plus fréquent dans le nouveau corpus (que dans le corpus de référence) et est en même temps relativement peu fréquent dans le corpus de référence.

Le Tableau 5 montre les scores de nouveauté obtenus pour nos données. Les topics avec un score positif sont les topics T1,

T1	T2	T3	T4	T5	T6
108.81	1.56	-0.96	19.14	-0.61	-0.88

TABLE 5: Scores de nouveauté pour les topics de notre corpus.

T2 et T4 pour lesquels nous avons constaté (manuellement) qu'ils correspondent au nouveau sens *geste* de « quenelle ».

5. Explication des calculs : Le corpus *REF* contient 164 paragraphes dans le sens *culinaire* auxquels nous pouvons ajouter 31 paragraphes dans le sens *culinaire* issus du corpus *NOUV*, suite à notre vérification manuelle. Le corpus *NOUV* contient 342 paragraphes dont 38 n'ont pas le sens *geste* : 31 véhiculent le sens *culinaire* et 7 ont un thème indéterminé.

5 Conclusion

Nous avons présenté une application d'une méthode à base de Topic Models pour l'identification du nouveau sens du signifiant « quenelle ». Les diverses mesures proposées par Lau *et al.* (2012, 2014) constituent des critères pertinents dans notre cas. Cette méthodologie reste à valider et à reconduire sur d'autres changements de sens connus comme « caviar » ou « souris ». L'un des points problématiques de la méthode réside dans l'alignement des topics avec les définitions trouvées dans les dictionnaires : les répertoires de sens et la granularité sont fortement dépendants du dictionnaire utilisé. Il semblerait donc plus réaliste de détecter automatiquement les évolutions thématiques et de laisser aux lexicographes le soin de les décrire sur la base des indices automatiques.

Remerciements

Ces travaux ont été financés par l'Université de Strasbourg dans le cadre de l'Initiative d'Excellence (IdEx) 2012 (projet Logoscope).

Références

BLEI D. M., NG A. Y. & JORDAN M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, **3**.

BOUSSIDAN A. & PLOUX S. (2011). Using topic salience and connotational drifts to detect candidates to semantic change. In *Proceedings of IWCS 2011*.

CABRÉ T. & NAZAR R. (2011). Towards a New Approach to the Study of Neology. In *Neology and Specialised Translation 4th Joint Seminar Organised by the CVC and Termisti*.

JI H., PLOUX S. & WEHRLI E. (2003). Lexical Knowledge Representation with Contexonyms. In *Proceedings of the 9th Machine Translation*, p. 194–201.

LAU J. H., COOK P., MCCARTHY D., GELLA S. & BALDWIN T. (2014). Learning Word Sense Distributions, Detecting Unattested Senses and Identifying Novel Senses Using Topic Models. In *Proceedings of ACL 2014*, Baltimore, USA.

LAU J. H., COOK P., MCCARTHY D., NEWMAN D. & BALDWIN T. (2012). Word sense induction for novel sense detection. In *Proceedings of EACL 2012*.

REUTENAUER C. (2012). *Vers un traitement automatique de la néosémie : approche textuelle et statistique*. PhD thesis, Université de Lorraine.

REUTENAUER C., JACQUEY E., LECOLLE M. & VALETTE M. (2010). Sémème au macroscope : genèse et variation sémiques d'une unité lexicale. In *Proceedings of 10th International Conference Journées d'Analyse statistique des Données Textuelles*, Sapienza University of Rome : Sergio Bolasco, Isabella Chiari, Luca Giuliano.

ROMARY L., SALMON-ALT S. & FRANCOPOULO G. (2004). Standards going concrete : from LMF to Morphalou. In M. ZOCK, Ed., *COLING 2004 Enhancing and using electronic dictionaries*, Geneva, Switzerland.

STEYVERS M. & GRIFFITHS T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, **427**(7).

TEH Y. W., JORDAN M. I., BEAL M. J. & BLEI D. M. (2006). Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, **101**(476).

Grammaire récursive non linéaire pour les langues des signes

Michael Filhol
LIMSI–CNRS, B. P. 133, 91403 Orsay cedex
michael.filhol@limsi.fr

Résumé. Cet article propose une approche pour la formalisation de grammaires pour les langues des signes, rendant compte de leurs particularités linguistiques. Comparable aux grammaires génératives en termes de récursivité productive, le système présente des propriétés nouvelles comme la multi-linéarité permettant la spécification simultanée des articulateurs. Basé sur l'analyse des liens entre formes produites/observées et fonctions linguistiques au sens large, on observe un décloisonnement des niveaux traditionnels de construction de la langue, inhérent à la méthodologie employée. Nous présentons un ensemble de règles trouvées suivant la démarche présentée et concluons avec une perspective intéressante en traduction automatique vers la langue des signes.

Abstract. This article presents a formal approach to Sign Language grammars, with the aim of capturing their specificities. The system is comparable to generative grammar in the sense that it is recursively productive, but it has quite different properties such as multilinearity, enabling simultaneous articulator specification. As it is based on the analysis of systematic links between observable form features and interpreted linguistic functions in the general sense, the traditionally separate linguistic levels all end up covered by the same model. We present the results found for a set of linguistic structures, following the presented methodology, and conclude with an interesting prospect in the field of text-to-Sign machine translation.

Mots-clés : Grammaire formelle, multi-linéarité, langue des signes.

Keywords: Formal grammar, multilinearity, Sign Language.

1 Introduction : langue des signes

La communauté s'est longtemps intéressée aux signes à un niveau lexical. Du côté de la linguistique : inventaires de signes pour l'élaboration de dictionnaires de vignettes (cf. fig. 1) ou représentation "paramétrique" de signes isolés (Stokoe, 1960), sorte de système phonologique selon lequel un signe est vu comme la combinaison de valeurs données à un ensemble prédéfini de paramètres (1 signe = 1 configuration + 1 orientation + 1 emplacement + 1 mouvement des mains). En TAL : conception de systèmes d'animation de signes placés en séquence (Hanke, 2002), par analogie assimilatrice aux séquences de mots des phrases écrites. À défaut de modèles plus adaptés, la langue des signes a peu été étudiée autrement que comme un alignement de signes manuels réputés lexicaux. Aujourd'hui, nous manquons de connaissances et (donc) de modèles formels capables de représenter l'organisation du discours signé.

FIGURE 1 – Vignette de dictionnaire LS pour le signe "parking" (Moody, 1997)

La difficulté est due à ce qu'en langue des signes, langue orale par excellence, beaucoup d'articulateurs simultanés contribuent à la construction de l'énoncé (sourcils, doigts, buste, ligne des épaules...), tous à des fréquences, niveaux et moments différents (cf. fig. 2). Aussi, la nature lexicale de certaines unités et leur compositionnalité phonologique ne

fait pas consensus chez les linguistes. La pensée assimilatrice et générativiste (Marshall & Sáfár, 2004) suppose que les niveaux linguistiques (phonétique, phonologique, morphologique, lexical, syntaxique, etc.) identifiés pour décrire les langues écrites et, déjà dans une moindre mesure, vocales, sont valides pour les LS sans quoi leur statut de langue pourrait être remis en cause. Au contraire, d'autres observent sur corpus de nombreux éléments pour lesquels il est impossible de trancher s'il s'agit d'éléments lexicaux ou de constructions syntaxiques, ou encore si un geste est linguistique ou "co-verbal", c'est-à-dire à considérer comme hors du système linguistique. Par exemple, l'expression de relations spatiales entre entités du discours se fait par un usage direct de l'espace de signation, respectant la topologie de la relation (Cuxac, 2000). Ce type de construction, fortement iconique [1] ne fait pas partie des lexiques car leur nombre est infini, mais n'est pas relégable au non-verbal. Doit-on y voir une construction syntaxique alors qu'elle ne comporte pas d'éléments lexicaux identifiés et que tout est simultané ?

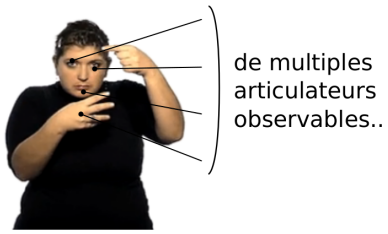


FIGURE 2 – LS : de multiples articulateurs simultanés

Aussi, l'emploi plus que récurrent de ces structures fortement iconiques en LS amène souvent à questionner les hypothèses traditionnellement admises liées à l'arbitraire du signe. Ces structures consistent en des unités non stabilisées, au sens où aucun inventaire ne les ferait apparaître comme des unités de construction, mais utilisant les propriétés du canal gestuel (trois dimensions, relations topologiques dans l'espace de signation, angles entre articulateurs et formes des mains) pour créer de nouvelles unités au fil du discours de manière extrêmement productive. Contrairement au mime, celles-ci sont pourtant bien linguistiquement régies et ne peuvent être improvisées. Si elles sont clairement efficaces pour les relations spatiales et la description de formes, l'utilisation symbolique de l'espace est tout aussi récurrente pour les concepts abstraits. L'iconicité fait donc véritablement partie intégrante de la langue et doit être selon nous prise en compte dans tout effort de modélisation des LS.

2 Une grammaire fonction–forme

Sans vouloir arbitrer le débat linguistique, nous avons ces dernières années proposé un modèle formel adapté à l'utilisation en informatique de ces langues naturelles, qui prend en compte leurs dimensions orale – même uniquement orale donc fondamentalement non linéaire, où plusieurs réalisations simultanées sont possibles au contraire de l'écrit – et iconique dans l'espace gestuel.

2.1 Forme, fonction et règles

Pour appréhender la LS sans influence de l'écrit, nous avons d'une part basé nos études sur des corpus de vidéos de langue attestée par des locuteurs natifs de LS, et fait le choix de revenir à une hypothèse linguistique de base, moins contraignante, selon laquelle la langue permet un lien entre :

– des formes produites, que l'on peut annoter dans un logiciel d'annotation en pistes (fig. 3), e.g. en LS la direction du regard, les mouvements des épaules, du buste, la durée d'une pause, les rotations de la tête, positions des mains, la flexion d'un doigt ;
– et leurs fonctions, interprétées à la réception du message, p. ex. dater un événement, séparer deux parties d'un discours, intensifier une fonction adverbiale, marquer l'objectivité d'une mesure...

L'idée est alors d'identifier, à partir de données réelles (corpus de langue), les liens systématiques qu'il peut exister, quitte à les paramétrer, entre forme et fonction, et de les formaliser. Toute fonction interprétée systématiquement pour un

1. L'*iconicité* est la propriété pour un signe, un signifiant ou toute forme linguistique produite de ressembler ou de rappeler le sens, le signifié, le concept associé.

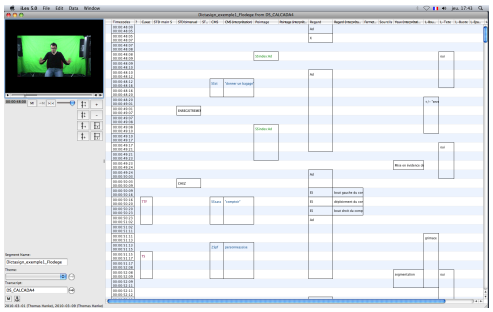


FIGURE 3 – Annotation en pistes (ici verticales avec le logiciel *iLex*)

critère de forme donné constitue pour nous une *règle d'interprétation* de la forme observée. Inversement, la description des invariants de forme dans les occurrences d'une fonction donnée permet la formalisation d'une *règle de production* pour cette fonction. Dans notre but particulier de synthèse de LS, ce sont ces dernières qui vont constituer la base d'une grammaire formelle de génération propre à la langue des signes observée, permettant d'animer un personnage virtuel en synthétisant les formes correspondant aux fonctions qui seraient données.

2.2 Méthodologie pour la recherche de règles

Pour élaborer les règles de production, la méthodologie utilisée vise à identifier les liens entre fonction et forme en opérant un va-et-vient entre les deux, associant à chaque fois des descriptions de formes (resp. fonctions interprétées) aux occurrences d'un critère de fonction (resp. de forme) :
– dans le sens de l'*observation*, repérer toutes les occurrences d'un critère fonctionnel et en décrire les formes (articulations et précédences temporelles) ;
– dans le sens de l'*interprétation*, repérer toutes les occurrences d'un critère de forme et en décrire la fonction (interprétation de l'intention sémantique, au sens large, du locuteur une fois le message compris).
Dans les deux cas, il s'agit de dégager les éléments communs dans les descriptions élaborées, et procéder au sens inverse à partir du critère ainsi dégagé. Le va-et-vient entre observation et interprétation permet d'affiner les critères de fonction jusqu'à converger vers un invariant paramétré de forme à produire pour une fonction de la langue.

Par exemple, à partir du critère de forme illustré par la figure 4 (un pointage de l'index d'une main sur l'autre main en configuration chiffrée), nous avons pu dérouler la recherche suivante.

1. Recherche en interprétation, critère de forme défini ci-dessus (fig. 4) :
   → ici, toutes les occurrences faisaient partie d'une liste énumérée.
2. Recherche en observation, critère fonctionnel "énumération" :
   → ici, un grand nombre d'occurrences ont fait apparaître le même mouvement de tête vers l'en avant, sur chaque item énuméré.
3. Recherche en interprétation, critère de forme "mouvement de tête vers l'avant" :
   → ici, toutes les occurrences marquent des éléments non mutuellement exclusifs d'énumérations non-exhaustives.
4. Recherche en observation, critère fonctionnel "énumération non-exhaustive d'items non mutuellement exclusifs" – nous appellerons ces énumérations "listes ouvertes" :
   → retrouvant la même liste qu'à l'étape précédente, on propose de s'arrêter directement ici ; le critère fonctionnel "liste ouverte" donne ici déjà systématiquement lieu à la même forme, ce qui permet une règle de production.

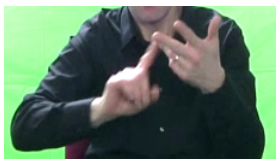


FIGURE 4 – Illustration d'une forme initiale

Accessoirement, on note une utilisation totalement variable de la forme initiale avec pointage de l'index dans les énumérations qui l'utilisent, tout à fait optionnelle et sans effet. Le critère de départ a donc disparu dans le processus itératif. De manière générale, si un choix intuitif au départ peut accélérer la convergence vers une règle, celui-ci n'est pas décisif. Aussi, l'exemple ici montre une entrée par la forme, mais l'entrée par la fonction est également possible, le processus conduisant à l'alternance des points de vue.

3 Résultats récents et validation de l'approche

3.1 Système de règles temporelles

Suivant la méthodologie décrite en partant de fonctions temporelles (événements en séquence, duratifs, etc.), nous avons déjà pu établir un système cohérent de huit règles stables dont les fonctions sont (Filhol *et al.*, 2013) :

(r1) Succession de deux événements séparés par une période de moins de 10 jours
Arguments : la durée *dur* (optionnelle), les événements *pre* et *post*

(r2) Séquence chronologique de dates, périodes/durées et événements
Arguments : N éléments $item_i$ dans l'ordre chronologique, $i \in [1..N]$

(r3) Période de 10 jours minimum
Arguments : la durée *dur* de la période

(r4) Un événement dure au moins 10 jours
Arguments : la durée *dur* et l'événement *event*

(r5) Un événement dure moins de 10 jours
Arguments : la durée *dur* et l'événement *event*

(r6) Événement daté
Arguments : la *date* et l'événement *event*

(r7) Un événement s'écoule du temps de l'énonciation à une borne temporelle
Arguments : la date de fin *d2* et l'événement *event*

(r8) Un événement s'écoule entre deux bornes temporelles dont la première est différente du temps de l'énonciation
Arguments : les bornes temporelles *d1* et *d2*, et l'événement *event*

Les formes associées sont ensuite décrites avec le formalisme "AZee", non développé ici mais présenté ailleurs (Filhol *et al.*, 2014). Ces formes peuvent être schématisées comme sur la figure 5, où :

– les rectangles délimitent sur l'axe horizontal les intervalles temporels durant lesquels la forme qu'ils contiennent est produite (l'unité de temps reste ici fictive mais des mesures par capture de mouvement nous permettront de préciser les spécifications sur cet axe) ;
– l'arrangement des rectangles sur l'axe vertical est arbitraire et n'est pas à interpréter comme les pistes d'une annotation où l'on trouverait un articulateur annoté par ligne ;
– les éléments en italique sont les noms des arguments des règles, en d'autres termes les boîtes à remplir ;
– les articulations décrites peuvent être de tout type et viser tout articulateur, ou faire appel à une autre règle, imbriquée, par exemple : '`hd:fwd`' = tête déplacée en avant, '`eg:s-sp`' = regard sur l'espace de signation, '`el:semi-cl`' = paupières semi-fermées, '`hd:nod-up`' = menton relevé, etc.

3.2 Discussion

Ce résultat est pour le moins intéressant, et ce pour trois raisons que nous présentons ci-dessous.

Compositionnalité récursive Nous avons produit un système de règles, chacune produisant la signation (forme) de la fonction portée, et pouvant comporter des dépendances (paramètres). Par exemple, la forme décrite par (r7) dépend de celle des éléments en italique, à savoir un événement et une date. Nous observons que ces signations peuvent elles-mêmes provenir d'une production obtenue par une entrée fonctionnelle. Par exemple, on trouve dans le corpus la phrase : *Dix ans après l'évacuation musclée de l'église Saint-Bernard, le 23 août 1996 à Paris, les sans papiers et leurs soutiens ne veulent pas être "dans la commémoration" mais dans le "combat", ...*

Celle-ci est traduite avec l'imbrication suivante, en parfaite cohérence avec les fonctions sémantiques des règles utilisées :

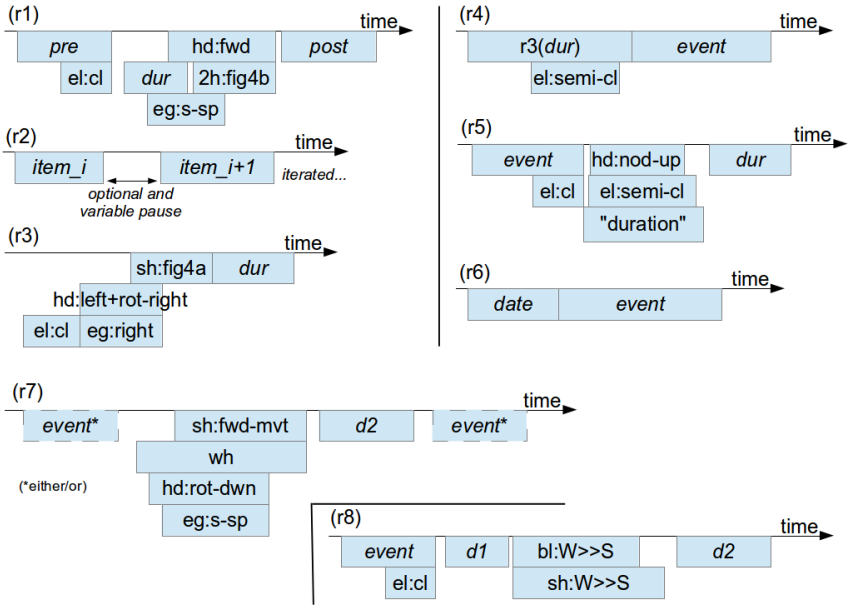


FIGURE 5 – Règles de production

```
r2(item_1 = r6(date = "23/08/1996",
               event = "évacuation musclée de l'église"),
   item_2 = r3(dur = "10 ans"),
   item_3 = ...)
```

Cette observation est importante car elle vérifie la propriété pour les règles d'être mutuellement récursives, condition sans laquelle le langage généré par la grammaire serait fini, donc la couverture du modèle nécessairement incomplète. Cette propriété est donc partagée avec les grammaires génératives, mais il reste une différence de taille avec celles-ci : l'ordre des feuilles ne représente absolument rien de la séquence finale du discours. En effet, celle-ci n'est pas vue comme une séquence, les règles pouvant mettre leur contenu en parallèle. Les sous-arbres ne sont finalement ordonnés que pour être nommés, et s'il s'agit de formes à signer, leur position relative temporelle dépend complètement de ce que dicte la règle.

Précision des règles Certaines fonctions peuvent paraître pour le moins étonnantes. Pour cause, notre méthodologie étudie la langue des signes à part et sans regard sur le texte, comme une langue nouvelle méritant ses formalismes propres. La distinction entre les périodes de plus et moins de 10 jours est un exemple de ce qu'aucun paradigme TAL syntaxico-sémantique n'aurait envisagé, et pourtant bien vérifié en corpus. Aussi bien dans le cas des durées d'événements que des durées de séparation entre événements consécutifs, on observe ce même seuil de durée. Notre approche rigoureusement basée sur les données nous semble éviter les biais de la "pensée textuelle" (linéarité, organisation syntaxique de mots prélistés...) et garantir que les règles de production trouvées soient bien les briques d'une "pensée signe".

Niveaux linguistiques des règles La vision traditionnelle en TAL, et jusqu'alors transposée en TALS, considère tout énoncé comme une séquence d'éléments lexicaux déterminés a priori, éventuellement modifiés par les règles syntaxiques gouvernant l'ordre de la séquence, et séparés par une période de transition plus ou moins courte. Cette vision a longtemps été portée par le courant générativiste et considérée comme universelle. Ici, les niveaux syntaxique, sémantique, etc. ne sont pas a priori délimités, et les opérations linguistiques – et donc les règles correspondantes – ne nécessitent pas d'être catégorisées en niveaux avant d'être décrites. Quels que soient leurs niveaux, un lien direct est créé entre ces fonctions et des articulations observables, assimilables en partie à un niveau phonétique, tout en imbriquant des structures de niveaux supérieurs. Notre approche décloisonne donc l'ensemble des fonctions linguistiques décrites, remettant ainsi en question l'universalité du modèle à couches (niveaux empilés). La validation de ce modèle multi-linéaire et sans dominance syntaxique, en passant à l'échelle d'une couverture plus large de la langue, permettra de nourrir la réflexion sur les universaux du langage.

4 Conclusion et perspective en traduction

Nous avons présenté une approche et une méthodologie pour élaborer des grammaires formelles pour les langues des signes, basées sur corpus et sans hypothèse linguistique quant aux niveaux et fonctions modélisées. Cette approche permet d'écrire les éléments de forme invariants, quitte à les paramétrer, pour des fonctions linguistiques identifiées, chacune donnant lieu à une règle de production directement animable par un signeur virtuel par exemple.

Une perspective intéressante d'utilisation du modèle existe selon nous en traduction automatique de texte vers LS. L'imbrication des règles représentant l'énoncé à signer est donnée en entrée du système de génération peut servir directement de représentation fonctionnelle du discours à traduire, puisque les règles contenues dans celle-ci portent, par construction, chacune une fonction interprétée (lexicale, sémantique, à but rhétorique, etc.). Avec une vision similaire à celle employée côté cible (liens forme–fonction), nous proposons de rechercher les formes du côté source ayant la même fonction. En d'autres termes et avec un point de vue TAL, pour chaque règle de production connue en langue des signes, il s'agit de repérer dans le texte les formes (en français écrit cette fois) pouvant être interprétées de la manière donnée par la fonction de la règle. Lorsqu'une forme textuelle est ainsi repérée, la règle peut alors être déclenchée pour une utilisation dans la traduction à proposer en sortie. Par exemple, la règle "liste ouverte", dont la fonction est l'énumération non-exhaustive d'items non mutuellement exclusifs, peut être déclenchée dès lors que l'on observe dans le texte une suite d'éléments séparés par des virgules et de même type syntaxique, et introduite par "comme" ou "par exemple".

Le problème global, illustré sur la figure 6, devient donc un ensemble de sous-problèmes TAL de recherche d'information, chacun prévu pour déclencher une règle en utilisant tous les moyens possibles (apprentissage, règles à base de motifs...), et ses arguments autant que possible. Les règles étant déclenchées séparément, elles doivent ensuite être combinées (nuage gris) pour former la sortie du système de traduction (cadre bleu), servant du même coup d'entrée du système de synthèse (cadre jaune). Nous démarrons aujourd'hui ce travail exploratoire et espérons en présenter des résultats prochainement.

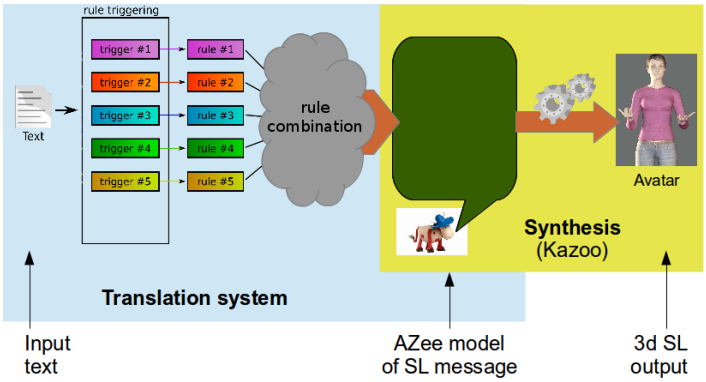


FIGURE 6 – Architecture pour une traduction automatique à base de déclenchement de règles

Références

CUXAC C. (2000). *Langue des signes française, les voies de l'iconicité*, volume 15–16. Paris Ophrys.

FILHOL M., HADJADJ M. N. & CHOISIER A. (2014). Non-manual features : the right to indifference. In *Language resource and evaluation conference (LREC), 6th workshop on the representation and processing of Sign Language : beyond the manual channel*, Reykjavik, Islande.

FILHOL M., HADJADJ M. N. & TESTU B. (2013). A rule triggering system for automatic text-to-sign-translation. In *Sign Language translation and avatar technology (SLTAT)*, Chicago, USA.

HANKE T. (2002). Visicast deliverable d5-1 : interface definitions. ViSiCAST project report.

MARSHALL I. & SÁFÁR E. (2004). Sign language generation in an ale hpsg. In *Proceedings of HPSG04*.

MOODY B. (1997). *La langue des signes, dictionnaire bilingue LSF/français*. Paris : IVT.

STOKOE W. (1960). Sign language structure : an outline of the visual communication system of the american deaf. *Studies in Linguistics, Occasional Papers*, **8**.

Vers un traitement automatique en soutien d'une linguistique exploratoire des Langues des Signes

Rémi DUBOT, Arturo CURIEL, Christophe COLLET
IRIT, Université de Toulouse, 118 route de Narbonne, 31062 Toulouse Cedex 9, France
prenom.nom@irit.fr

Résumé. Les langues des signes sont les langues naturelles utilisées dans les communautés sourdes. Elles ont suscitées, ces dernières années, de l'intérêt dans le domaine du traitement automatique des langues naturelles. Néanmoins, il y a un manque général de données de terrain homogènes. Notre réflexion porte sur les moyens de soutenir la maturation des modèles linguistiques avant d'entamer un large effort d'annotation. Cet article présente des pré-requis pour la réalisation d'outils s'inscrivant dans cette démarche. L'exposé est illustré avec deux outils développés pour les langues des signes : le premier utilise une logique adaptée pour la représentation de modèles phonologiques et le second utilise des grammaires formelles pour la représentation de modèles syntaxiques.

Abstract. Sign Languages (SLs) are the vernaculars of deaf communities, they have drawn interests in the natural language processing research in recent years. However, the field suffers a general lack of homogeneous information. Our reflexion is about how to support the maturation of models before starting a consequent annotation effort. In this paper, we describe the requirements of tools supporting such an approach. It is illustrated with two examples of work developed following these guidelines. The first one aims at phonological representation using logic and the second one targets supra-lexical features recognition.

Mots-clés : Langues des Signes, Formalismes de modélisation, Reconnaissance, Annotation semi-automatique .

Keywords: Sign Languages, Modeling formalisms, Recognition, semi-automatic annotation.

1 Introduction

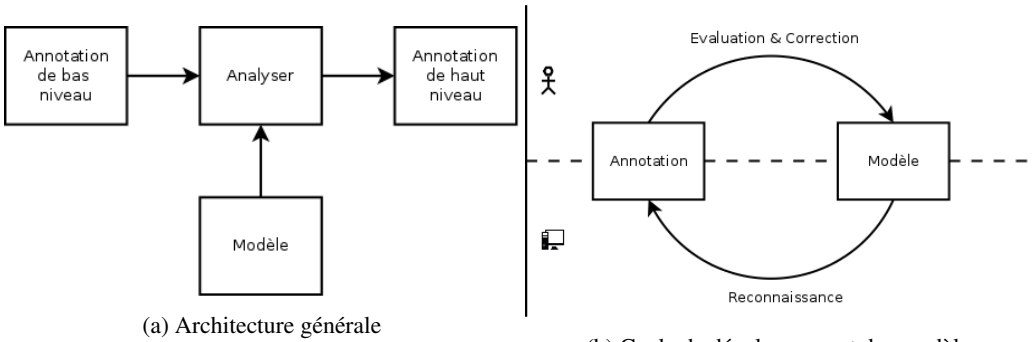
Les Langues des Signes (LS) sont les langues naturelles utilisées par les personnes sourdes comme moyen de communication principal (Stokoe, 2005). Elles se caractérisent par leur usage de l'espace en conjonction avec un ensemble de parties du corps –les articulateurs– pour véhiculer de l'information (Valli & Lucas, 2000). Bien que les mains constituent l'articulateur le plus saillant, il ne doit pas occulter le rôle essentiel de la Gestuelle Non-Manuelle (GNM) en général et de la gestuelle faciale en particulier (Cuxac, 2000). Cette multiplicité des articulateurs associée à leurs caractéristiques propres confèrent aux LS une dimension multi-linéaire non décrite pour les langues vocales. Il est maintenant bien établi que ces langues ont un pouvoir d'expression équivalent aux langues vocales.

L'intérêt de l'étude des LS dépasse largement le cadre des communautés de locuteurs. Dans le domaine linguistique, l'étude des LS contribue à l'amélioration de la compréhension des mécanismes sous-jacents de la communication humaine par la recherche de caractéristiques communes entre langues vocales et LS (Emmorey, 2001). L'intérêt croissant pour ce domaine a soulevé des questions autour de l'obtention, la comparaison et l'évaluation des données (Baker *et al.*, 2008). Les données sont toujours un point sensible en linguistique, mais l'absence de forme écrite (d'usage courant) complique encore la tâche pour les LS. Les corpus prennent la forme de documents vidéo et d'éventuelles données de capture de mouvement, accompagnés de méta-données. Une forme qui réclame des solutions de capture, de stockage et d'analyse spécifiques (Schembri *et al.*, 2014). Ces contraintes impactent lourdement les coûts de création et maintenance des corpus et par conséquence leurs tailles et leurs disponibilités.

Les problèmes posés par le support pour la création des corpus se répercutent sur l'annotation. Le processus d'annotation, bien que fondamental, est long et fastidieux (Neidle *et al.*, 2001) ; il pose des problèmes de reproductibilité des résultats, aggravé par le manque d'annotateurs (Brugman *et al.*, 2004). Même dans le meilleur des cas, la tâche n'est pas simple, les paramètres d'intérêt varient, sont nombreux et leur codage n'est pas standardisé. Tout cela motive le développement d'outils d'annotation automatique qui permettent de diminuer l'impact de ces difficultés (Dreuw *et al.*, 2010).

L'annotation automatique des LS est fortement liée à la reconnaissance de formes. Plusieurs outils ont été créés pour la détection et le suivi des articulateurs sur les images. On a d'abord vu des approches invasives utilisant des capteurs portés par le signeur comme des gants (Mehdi & Khan, 2002). Puis, avec les progrès techniques, des approches non-invasives sont apparues comme du suivi des mains et du visage dans les vidéos (Gonzalez & Collet, 2012) ou du suivi 3D avec de nouveaux capteurs (Zafrulla *et al.*, 2011). Enfin, nous sommes à un point où les solutions de reconnaissance emploient des approches très différentes, ce qui bloque une évaluation commune (Zahedi *et al.*, 2006).

Avec encore de nombreux aspects des LS mal compris et modélisés, il y a nécessité de créer des outils d'aide pour la recherche linguistique, ouverts à une démarche exploratoire. En effet, la démarche commune actuelle consiste à recourir fortement aux solutions d'apprentissage automatique développées pour les langues vocales. Outre leur adéquation sujette à débat, c'est surtout leur nécessité de large quantités de données d'apprentissage qui pose problème ici. Les démonstrations actuelles impliquent des gros efforts d'annotation spécifique pour de très petit modèles. Cela ne laisse guère de place à l'expérimentation sur les bases théoriques. En effet, les modèles appris sont fortement liés à la théorie sous-jacente à l'annotation (Zaenen, 2006). Or, étant donnée la masse de données en jeux, il n'est pas envisageable de répercuter sur toutes les annotations chaque modification de la théorie. Dans cet article, nous introduisons une nouvelle approche mettant l'accent sur l'identification du modèle (et sa formalisation) et sur l'adaptabilité à des annotations en faibles quantités et hétérogènes. Dans l'architecture adoptée, un analyseur générique utilise un modèle qui lui est donné pour compléter une annotation par la compréhension qu'il en a (cf. figure 1a). L'annotation en entré peut donc provenir d'autre outils de reconnaissance ou d'une annotation manuelle. Enfin, une évaluation de l'annotation en sortie de l'analyseur permet de corriger le modèle et donc des cycles de raffinage successifs (cf. figure 1b). Ce sont deux formalismes de représentation de modèles et deux analyseurs qui sont présentés : l'un au niveau phonologique (section 2) et l'autre au niveau syntaxique (section 3).



(a) Architecture générale

(b) Cycle de développement des modèles

FIGURE 1

2 Représentation sous-lexicale de la LS avec une logique formelle

La génération d'annotations lexicales pour les corpus est basée principalement sur la reconnaissance automatique de signes et d'autres structures. Actuellement, il existe différents outils automatiques capables de reconnaître des signes des LS, mais qui ne sont utilisables que dans des conditions très spécifiques. De ce fait, il est difficile de généraliser ces méthodes aux autres scénarios présents dans des tâches de Traitement Automatique de Langues Naturelles (TALN). Pourtant, des outils génériques sont nécessaires pour améliorer l'étendue de l'annotation des corpus et donc la quantité d'information disponible pour la reconnaissance.

Pour arriver à comprendre la LS, y compris au niveau lexical, il est nécessaire de suivre les nombreuses parties du corps du signeur (nommés *articulateurs*) qui participent à la transmission de l'information langagière. Ainsi, chacun de ces articulateurs agit comme un canal de communication à part. Pour cette raison, il est commun d'avoir des outils de capture très différents pour obtenir les données de chacun des canaux (les mains, les gestuelle non manuelle (GNM), etc). Cela montre qu'un système de reconnaissance doit être capable d'intégrer plusieurs types de sources d'informations simultanées, spécialement si son objectif est d'aller vers la reconnaissance supra-lexicale. Cependant, la diversité des représentations de signes que les systèmes existants adoptent compliquent une telle intégration. Tout cela oblige à réfléchir sur la manière de représenter les LS dans les tâches de reconnaissance, même au plus bas niveau.

Des travaux comme ceux de (Vogler & Metaxas, 2001) résolvent quelques uns de ces problèmes avec l'adoption d'une *segmentation phonologique* des signes. Ils se basent sur des modèles linguistiques comme ceux présentés par (Liddell & Johnson, 1989) et (Sandler, 1989). Dans leur système, les signes sont séparés en deux types d'intervalles temporels : intervalles de changement (*mouvements*) et intervalles statiques (*postures fixes*). En plus, les auteurs sont capables de réduire la complexité de la reconnaissance en traitant chaque canal de communication de manière individuelle, en parallèle. Pourtant, leur *framework* n'est pas facilement adaptable pour analyser des corpus sans données d'entrainement.

Par ailleurs, dans le domaine de la synthèse des LS, des auteurs comme (Losson & Vannobel, 1998) et (Filhol, 2008) ont introduit des spécifications formelles pour la description des signes. Cependant, ces formalismes sont difficiles à utiliser pour l'annotation, car ils se basent sur la construction de descriptions fondées sur des contraintes géométriques. Par conséquent, un système d'annotation basé sur ces descriptions doit évaluer des paramètres géométriques très fins, difficilement détectables de manière automatique.

2.1 La logique propositionnelle dynamique pour la langue de signes

Les auteurs (Curiel & Collet, 2013) ont défini une logique modale, la logique propositionnelle dynamique pour la LS (PDL_{SL}), pour la description formelle de la phonologie des LS. Avec cette logique, l'information phonologique de la LS peut-être représentée au travers d'énoncés atomiques. Ainsi, il est possible de modéliser des signes et d'autres structures comme des formules.

La PDL_{SL} permet également d'interpréter les corpus vidéos comme des systèmes de transition d'états (STE), où chacun des états correspond à une posture fixe et chacune des transitions correspond à un mouvement. Ces STE servent à faire de la vérification logique pour savoir si une certaine propriété des LS – ou un certain signe – existe dans une vidéo. Un exemple de ce processus est montré dans la figure 2.

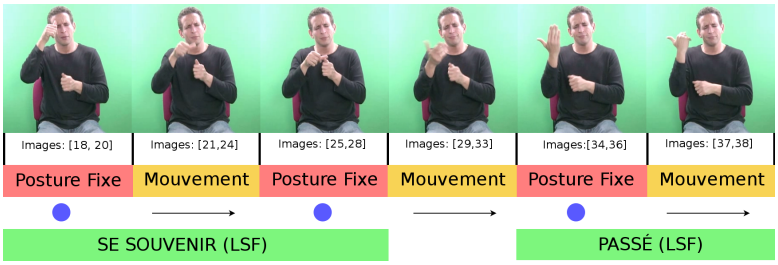


FIGURE 2 – Exemple des différentes couches présentes dans un système de reconnaissance basé sur la PDL_{SL}

L'intérêt principale d'utiliser la PDL_{SL} est d'avoir un langage formel pour décrire les actions parallèles des articulateurs. Dans ce formalisme, chacun des articulateurs peut être pris comme un agent indépendant qui agit sur une posture fixe. Cette logique permet d'exprimer l'effet des actions sur la posture de départ. Tout cela rends possible la construction de règles phonologiques de base capables d'améliorer la reconnaissance lexicale, y compris avec un manque d'information.

Le formalisme utilise le lambda calcul pour représenter de l'information incomplète. De cette manière, les informations non détectables peuvent être incluses dans les formules à travers des variables métasyntaxiques. Par exemple, les auteurs (Gonzalez & Collet, 2012) font la segmentation de postures fixes à partir de la vitesse des mains. Leur outil de suivi ne permet pas de reconnaitre les configurations des mains. Cependant, ils utilisent une mesure de proximité pour arriver à identifier si une configuration a changé. Sous le même principe, il est possible de savoir si deux mains ont la même configuration sans reconnaître exactement laquelle. La figure 3 montre une formule pour représenter cette situation. La même formule peut être utilisée aussi avec des systèmes qui arrivent à reconnaître les configurations.

$$même\_config\_deux\_mains = \lambda c.(config\_main(\texttt{droite}, c) \wedge config\_main(\texttt{gauche}, c))$$

FIGURE 3 – Formule qui représente un signe où les deux mains ont la même configuration c

2.2 Implémentation et évaluation d'un système de reconnaissance basé sur la $\mathrm{PDL_{SL}}$

Une première implémentation d'un système basé sur la $\mathrm{PDL_{SL}}$ a été développé par (Curiel & Collet, 2014). L'idée est que les utilisateurs puissent définir leurs propres bases de formules $\mathrm{PDL_{SL}}$, où chacune des formules décrit une structure lexicale d'intérêt. La solution inclut les outils nécessaires pour vérifier que les formules sont bien formées. En plus, le système intègre un solveur $\mathrm{PDL_{SL}}$, ce qui permet de trouver où chaque formule est satisfaite dans un STE. Le système génère chaque STE à partir des données de suivi d'un corpus. Chaque formule satisfaite dans un STE, représente une propriété contenue dans le corpus d'origine.

Cette version du logiciel est focalisée sur l'intégration de différentes technologies de suivi dans une même solution de reconnaissance. Il est possible de substituer les modules définis par le système avec des implémentations adaptées à d'autres outils de suivi. Cela permet de créer des modèles avec plus d'information, ce qui peut contribuer à améliorer le taux de reconnaissance. Ainsi, par exemple, le système peut être adapté pour travailler sur des données 2D ou 3D, juste en changeant de module. Pour cela, le système définit des interfaces d'implémentation. D'autre part, le même mécanisme permet d'intégrer d'autres modules de plus haut niveau pour raffiner les résultats de chaque étape, soit par des algorithmes d'apprentissage automatique ou par l'évaluation par des experts humains.

Une base de formule $\mathrm{PDL_{SL}}$ a été écrite pour permettre l'évaluation. La sortie du système a ensuite été comparée à une vérité terrain (manuellement annotée) pour produire les résultats présentés dans (Curiel & Collet, 2014).

3 Représentation de modèles syntaxiques avec des grammaires contraintes

Les linguistes montrent un intérêt croissant pour la GNM et en particulier son usage dans la syntaxe des LS. Les mécanismes de synchronisation des flux parallèles (les deux flux manuels et de nombreux non-manuels) constituent un sujet majeur dans l'étude du rôle de la GNM dans la syntaxe (Vermeerbergen *et al.*, 2007). Cette analyse a motivé le développement d'une solution pour l'annotation semi-automatique de structures syntaxiques (Dubot & Collet, 2014a).

Le formalisme de représentation de modèles présenté dérive des grammaires génératives. Il reprend l'idée introduite par Karlsson d'utiliser des contraintes pour exprimer les relations fines entre les unités (Karlsson, 1990). Il en abandonne cependant tous les détails qui rendent les travaux de Karlsson –et les travaux subséquents– spécifiques aux formes écrites des langues vocales. Le présent formalisme remplace la séquence du membre droit des règles de production par des ensembles d'unités en retirant ainsi l'information pseudo-temporelle. De tels types de grammaires sans séquences ont déjà été utilisés, au moins pour la compréhension d'images (Zhu & Mumford, 2006). L'information sur l'organisation temporelle est mise au même rang que celle pour la spatialisation ou l'articulation par exemple. De la même façon que pour les grammaires de traits, les unités sont dotées d'attributs comme, par exemple, un début et une fin dans le temps, une position dans l'espace, etc. L'organisation –temporelle et autre– est décrite par des contraintes entre les unités qui affectent les valeurs de leurs attributs. Pour restreindre le moins possible le champ des modèles représentables, la solution est compatible tant avec les grammaires lexicalisées, non-lexicalisées et mixant les deux paradigmes. Cela est rendu possible en abandonnant le concept de terminalité des symboles et en introduisant celui de détectabilité.

Le choix des grammaires génératives comme base introduit deux restrictions sur les modèles qui sont généralement considérées comme des obstacles. Premièrement, cette famille de formalismes nécessitent généralement une racine (couramment appelé axiome de la grammaire et noté S). Selon le cadre théorique dans lequel il s'ancre, un modèle peut présenter naturellement une telle racine, mais lorsqu'il s'agit d'expérimenter autour d'un phénomène syntaxique, le modèle est rarement complet et l'introduction d'une racine peut être artificielle ou même techniquement difficile. Ce problème se présente par exemple dans le cas de la modélisation par des grammaires lexicalisées (donc dont toutes les unités sont détectables) : une grammaire ne peut pas avoir une racine unique et lexicalisée, cela signifierait qu'il existe une unité gestuelle présente constamment. Ce problème est, en partie, résolu par la définition d'un ensemble de nœuds racines tel que présenté dans (Dubot & Collet, 2014a). La deuxième restriction posée par l'usage des grammaires génératives est la nécessité pour les modèles d'être complets, en particulier ceux-ci doivent descendre jusqu'au niveau des données en entrée du système de reconnaissance automatique. Par exemple, la personne voulant travailler sur les liens entre propositions d'un énoncé devra décrire dans son modèle toute la construction d'une proposition sans que ces détails soient nécessaires à son étude. La solution proposée pour ce problème est l'usage de modèles dits *boite-noires*. Ceux-ci se définissent par le fait qu'ils produisent des résultats montrant des caractéristiques globales similaires à ceux qu'un modèle classique aurait donné mais dont les détails ne sont pas interprétables. Concrètement, cela signifie que l'utilisation du modèle produit un arbre syntaxique dont la racine présente les valeurs souhaitées sur ses attributs. Par contre les nœuds sous-jacents peuvent

être inintelligibles. En reprenant l'exemple précédent, ce modèle sera capable de trouver les propositions avec les bonnes valeurs pour les attributs mais dont les constructions ne sont pas utilisables (mais aussi inutiles). Pour construire de tels modèles *boite-noires*, nous avons utilisé une forme simple de grammaires de dépendances. Les dépendances ont l'avantage d'être aisément annotables à partir des gloses [1], qui sont l'annotation la plus répandue. De plus, les résultats produits à l'aide de ces modèles sont des arbres, ce qui rend leur intégration à la solution aisée. L'intégration des dépendances au formalisme est présentée dans (Dubot & Collet, 2014b).

(a) Model (b) Result

FIGURE 4 – Exemple de modèle syntaxique

La figure 4a présente un modèle avec une seule règle de production. Il représente une description simple d'une question. Les contraintes y apparaissent comme des flèches rouges, les unités détectables comme des nœuds rouges et les unités associées à un modèle *boite-noire* en noir. La figure 4b montre un exemple de résultat de la reconnaissance automatique avec les attributs valués de façon à satisfaire les contraintes.

Parce qu'elle est conçue pour l'expérimentation d'hypothèses linguistiques, cette solution ne peut être optimisée pour une théorie linguistique en particulier. Chacun des points de divergence avec les formes traditionnelles de grammaires génératives va dans le sens de relâcher les contraintes. Ceci amène à en augmenter la complexité algorithmique. La capacité de l'analyseur à donner une solution en un temps raisonnable a été évaluée de façon empirique sur un corpus de synthèse. Les résultats montrent que l'outil (dont le code n'est pas optimisé) obtient un rappel de 20 à 80 % avec un temps de recherche de l'ordre de la seconde pour des modèles de –au moins– $\sim$ 20 unités et $\sim$ 60 règles (Dubot & Collet, 2014a). Toutefois, cette solution, en permettant de formaliser et valider des hypothèses linguistiques est en mesure d'aider à établir une base théorique suffisamment solide pour bâtir un outil d'analyse spécifique et optimisé.

4 Conclusion

Dans la situation actuelle, le manque et l'hétérogénéité des annotations est problématique. La reconnaissance automatique des LS requerra nécessairement un gros effort d'annotation à terme, en particulier pour devenir utilisable en production. Cependant, une annotation s'ancre toujours dans un cadre théorique. Le bénéfice à tirer d'une annotation de masse est donc directement tributaire de la maturité de la théorie sous-jacente. Ainsi, la qualité mais surtout la diversité des approches théoriques explorées est capitale. Pour cette raison, la position adoptée par les auteurs est celle de créer des outils dédiés à cette phase exploratoire.

Le cadre d'application visé amène à un cahier des charges particulier. Les travaux présentés dans cet article se concentrent sur la capacité à travailler avec des modèles linguistiques expérimentaux variés, et sur des données très diverses et potentiellement faibles en quantités. Par exemple, la PDL_{SL} peut être utilisée par des chercheurs en TALN pour construire et modifier rapidement des modèles lexicaux. Son usage peut aller plus loin en adaptant chaque modèle à des corpus différents, ce qui ouvre l'accès à de nouvelles sources d'information. Au niveau de l'analyse syntaxique, c'est la facilité de formalisation des modèles qui est ciblée, pour aider à la réalisation de modèles d'une partie de la langue. De cette façon, les utilisateurs sont libres de tester leurs hypothèses linguistiques.

Enfin, nous pensons que les résultats issus de la recherche sur le traitement automatique des LS sont généralisables à beaucoup de phénomènes impliquant des gestes et de la simultanéité. Par exemple, cela pourrait bénéficier à l'étude de l'intégration de la gestuelle co-verbale et de la prosodie au traitement des langues vocales.

1. En LS, les gloses sont des mots-clés représentant des signes lexicalisés.

Références

BAKER A., VAN DEN BOGAERDE B. & WOLL B. (2008). Methods and procedures in sign language acquisition studies. *A. Baker & B. Woll. Sign Language Acquisition. Amsterdam & Philadelphia : John Benjamins*, p. 1–49.

BRUGMAN H., CRASBORN O. & RUSSEL A. (2004). Collaborative annotation of sign language data with peer-to-peer technology. In *LREC, Lisbon*.

CURIEL A. & COLLET C. (2013). Sign language lexical recognition with propositional dynamic logic. In *ACL*, volume 2.

CURIEL A. & COLLET C. (2014). Implementation of an automatic sign language lexical annotation framework based on propositional dynamic logic. In *LREC Wkshp : Represent. and Process. of Sign Languages*, Reykjavic.

CUXAC C. (2000). *La langue des signes française (LSF) : les voies de l'iconocité*. Ophrys.

DREUW P., NEY H., MARTINEZ G., CRASBORN O., PIATER J., MOYA J. M. & WHEATLEY M. (2010). The SignSpeak project - bridging the gap between signers and speakers. In N. C. C. CHAIR), K. CHOUKRI & *et. al.*, Eds., *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta : European Language Resources Association (ELRA).

DUBOT R. & COLLET C. (2014a). A hybrid formalism to parse sign languages. *arXiv :1403.4467 [cs]*.

DUBOT R. & COLLET C. (2014b). Sign language gibberish for syntactic parsing evaluation. *arXiv :1403.4473 [cs]*.

EMMOREY K. (2001). *Language, cognition, and the brain : Insights from sign language research*. Psychology Press.

FILHOL M. (2008). *Modèle descriptif des signes pour un traitement automatique des langues des signes*. PhD thesis, Université Paris-sud (Paris 11).

GONZALEZ M. & COLLET C. (2012). Sign segmentation using dynamics and hand configuration for semi-automatic annotation of sign language corpora. In *Gesture and Sign Language in Human-Computer Interaction and Embodied Communication*, LNCS. Springer.

KARLSSON F. (1990). Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th conference on Computational linguistics-Volume 3*, p. 168–173 : Association for Computational Linguistics.

LIDDELL S. K. & JOHNSON R. E. (1989). *American sign language : The phonological base*. Gallaudet University Press, Washington. DC.

LOSSON O. & VANNOBEL J.-M. (1998). Sign language formal description and synthesis. *International Journal of Virtual Reality*, **3**, 27—34.

MEHDI S. & KHAN Y. (2002). Sign language recognition using sensor gloves. In *ICONIP*.

NEIDLE C., SCLAROFF S. & ATHITSOS V. (2001). Signstream : A tool for linguistic and computer vision research on visual-gestural language data. *Behav. Res. Meth. Instr.*, **33**(3), 311–320.

SANDLER W. (1989). *Phonological representation of the sign : Linearity and nonlinearity in American Sign Language*, volume 32. Walter de Gruyter.

SCHEMBRI A., FENLON J., JOHNSTON T. & CORMIER K. (2014). Documentary and corpus approaches to sign language research. *The Blackwell guide to research methods in sign language studies*.

STOKOE W. C. (2005). Sign language structure : An outline of the visual communication systems of the american deaf. *Journal of Deaf Studies and Deaf Education*, **10**, 3–37.

VALLI C. & LUCAS C. (2000). *Linguistics of American Sign Language Text, 3rd Edition : An Introduction*. Gallaudet University Press.

VERMEERBERGEN M., LEESON L. & CRASBORN O. A. (2007). *Simultaneity in Signed Languages : Form and Function*. John Benjamins Publishing.

VOGLER C. & METAXAS D. (2001). A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding*, **81**(3), 358–384.

ZAENEN A. (2006). Mark-up barking up the wrong tree. *Comput. Linguist.*, **32**(4), 577–580.

ZAFRULLA Z., BRASHEAR H., STARNER T., HAMILTON H. & PRESTI P. (2011). American sign language recognition with the kinect. In *ICMI*.

ZAHEDI M., DREUW P., RYBACH D., DESELAERS T. & NEY H. (2006). Continuous sign language recognition-approaches from speech recognition and available data resources. In *LREC Wkshp : Represent. and Process. of Sign Languages*, p. 21–24.

ZHU S.-C. & MUMFORD D. (2006). A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, **2**(4), 259–362.

Résumé Automatique Multilingue
Expérimentations sur l'Anglais, l'Arabe et le Français

Houda Oufaida[1] Omar Nouali[2] Philippe Blache[3]

(1) Ecole Nationale Supérieure d'Informatique ESI, BP 68M Oued Smar, 16270, El Harrach Alger Algérie
(2) Centre de Recherche sur l'Information Scientifique et Technique CERIST, Rue Des 3 Frères Aissou, Ben Aknoun Alger Algérie
(3) LPL, AMU, CNRS, 5 avenue Pasteur, 13100 Aix-en-Provence
h_oufaida@esi.dz, onouali@mail.cerist.dz, blache@lpl-aix.fr

Résumé. La tâche du résumé multilingue vise à concevoir des systèmes de résumé très peu dépendants de la langue. L'approche par extraction est au cœur de ces systèmes, elle permet à l'aide de méthodes statistiques de sélectionner les phrases les plus pertinentes dans la limite de la taille du résumé. Dans cet article, nous proposons une approche de résumé multilingue, elle extrait les phrases dont les termes sont des plus discriminants. De plus, nous étudions l'impact des différents traitements linguistiques de base : le découpage en phrases, l'analyse lexicale, le filtrage des mots vides et la racinisation sur la couverture ainsi que la notation des phrases. Nous évaluons les performances de notre approche dans un contexte multilingue : l'anglais, l'arabe et le français en utilisant le jeu de données TAC MultiLing 2011.

Abstract. The task of multilingual summarization aims to design free-from language systems. Extractive methods are in the core of multilingual summarization systems. In this paper, we discuss the influence of various basic NLP tasks: sentence splitting, tokenization, stop words removal and stemming on sentence scoring and summaries' coverage. Hence, we propose a statistical method which extracts most relevant sentences on the basis of their terms discriminant power. We conduct several experimentations in a multilingual context: English, Arabic and French using the TAC MultiLing 2011 dataset.

Mots-clés : Résumé multilingue, analyse discriminante, TAL, évaluation multilingue.
Keywords: Multilingual summarization, Discriminant analysis, NLP, Multilingual evaluation.

1 Introduction

L'expansion du réseau internet à travers le monde a rendu disponible des documents écrits en différentes langues. En effet, le contenu web en anglais recule devant l'essor des autres langues ; Wikipédia.fr compte plus de un million et demi d'articles en mai 2014. Le contenu multilingue est donc en constante évolution et gagne de plus de plus de place sur le web. De ce fait, le développement d'outils multilingue est devenu indispensable : le résumé automatique multilingue en est un exemple.

Le but du résumé automatique est de produire une version condensée d'un ou plusieurs textes en utilisant des techniques informatiques. Ceci aidera le lecteur à décider si un document contient l'information désirée en un minimum de temps et d'effort. Récemment, de nouvelles tâches sont apparues et ont donné « un coup d'air frais » au domaine. Parmi ces tâches, on retrouve le résumé multilingue. Porté par les ateliers TAC MultiLing 2011[1] et ACL MultiLing 2013[2], le résumé multilingue connaît une belle avancée grâce à la mise à disposition de corpus d'évaluation.

Cette tendance vient à l'encontre des premiers systèmes de résumé où l'idée était d'utiliser le traitement automatique des langues (TAL) dans le but de *reformuler* l'essentiel du texte source et de *générer* de nouvelles phrases:

[1] http://www.nist.gov/tac/2011/Summarization/
[2] http://multiling.iit.demokritos.gr/pages/view/662/multiling-2013

l'identification des paraphrases et la fusion d'informations en est un exemple (Barzilay, McKeown, 2005). Dans des cas plus rares, les techniques du TAL ont été utilisées pour produire des extraits tels que l'usage de l'analyse rhétorique RST (Rhetorical Structure Theory)(Marcu, 1998). Cependant, ces techniques requièrent des traitements de haut niveau souvent basés sur les ressources limitées et dépendantes à la langue.

Récemment, les approches statistiques ont prouvé leurs performances grâce à leur robustesse d'une part et à cause du manque d'outils TAL efficaces notamment dans un contexte multilingue d'une autre part. C'est dans ce cadre que s'inscrit notre proposition, notre système utilise un minimum de ressources linguistiques et est de ce fait facilement portable vers d'autres langues. Le processus d'extraction de phrases est purement statistique et repose sur les termes les plus discriminants. Nous utilisons ainsi une méthode d'analyse discriminante, mRMR (minimum Redundancy - Maximum Relevance) (Peng et al., 2005), pour la pondération des termes. Nous appliquons notre système sur trois langues : l'anglais, l'arabe et le français avec différents niveaux de traitements linguistiques: découpage en phrases, analyse lexicale, traitement des mots vides et racinisation.

Le reste de l'article est organisé comme suit : nous présentons dans la section 2 les principaux travaux dans le contexte multilingue notamment ceux des ateliers TAC Multiling 2011 et ACL Multiling 2013. Dans la section 3, nous décrivons notre méthode de pondération des termes et d'extraction de phrases. Nous discutons également les traitements linguistiques requis par notre système ainsi que leur niveau de dépendance à la langue. Dans la section 4, nous évaluons notre approche en utilisant un corpus multilingue et nous discutons les résultats obtenus. Enfin, nous terminons notre étude par une conclusion et quelques perspectives de recherche dans la section 5.

2 Travaux précédents

Le résumé multilingue consiste à concevoir des systèmes capables de résumer des documents écrits en différentes langues. Par conséquent, le système doit utiliser des techniques très peu dépendantes ou *idéalement* indépendantes de la langue source. A notre connaissance, (Mihalcea, Tarau, 2005) est le premier travail dans la thématique. Le système, TextRank, construit en premier lieu représentation graphique du texte où les nœuds représentent les phrases et les liens représentent la similarité entre ces phrases. Les auteurs définissent cette similarité, simplement, comme le nombre de tokens en commun. En deuxième lieu, les auteurs utilisent le principe de recommandation entre phrases en appliquant les deux algorithmes : PageRank et HITS. Ce travail a montré que l'utilisation d'un algorithme performant pour l'extraction de phrases avec un minimum de traitements liés à la langue peut conduire à des performances comparables à ceux utilisant des traitements linguistiques de haut niveau. Dans un contexte monolingue, (Ledeneva, 2008) affirme que les différents traitements linguistiques de base n'engendrent aucune amélioration des scores ROUGE (Lin, 2004) des résumés système en utilisant MFS (Maximal Frequent Sequences).

Dans un contexte multilingue, les tâches de base à savoir le découpage en phrases et la segmentation de ces phrases en un ensemble de tokens deviennent plus complexes. Jusqu'à présent, les chercheurs ont choisit d'appliquer diverses solutions, simplistes pour la plupart. (Litvak et al., 2010) utilisent une représentation vectorielle des documents et tentent de trouver la combinaison linéaire optimale entre 31 méthodes de notation de phrases. Les auteurs évaluent leur système sur deux langues : l'anglais et l'hébreu et utilisent un outil de découpage en phrases propre à chaque langue. (Boudin, Torres-Moreno, 2009) filtrent les mots vides et combinent entre la similarité cosinus et la mesure LCS (Longest Common Substring) basée sur les caractères pour la pondération des phrases. Les auteurs affirment que pour un seuil de similarité LCS supérieur à 0,6, le traitement remplace avantageusement la lemmatisation.

L'atelier TAC MultiLing 2011évalue les difficultés liées à l'adaptation des méthodes au contexte multilingue. En effet, le corpus d'évaluation est un corpus parallèle de 7 langues. (Conroy et al., 2011) distinguent deux types de langues : simple casse et deux casses et utilisent un système de découpage en phrases adapté à chaque type: FASST-ONE et FASST-CAP. (Das, Srihari, 2011) utilisent le séparateur standard « . » tandis que (Hmida, Favre, 2011) utilisent le dernier caractère du document. (Saggion, 2011) utilisent les APIs GATE disponibles pour 4 langues. L'atelier ACL MultiLing 2013 reprend la tâche de l'atelier TAC MultiLing 2011, à savoir « Résumé multi-documents multilingue » et intègre trois nouvelles langues dont le Chinois. Pour l'analyse lexicale, (Conroy et al., 2013) distinguent 3 types de langues : anglais, non anglais et langues idéographiques (chinois), ils utilisent pour chacune une expression régulière particulière.

Le découpage en phrases et l'analyse lexicale sont des taches de base du TAL et on peut soit utiliser des outils dédiés soit opter pour des solutions agressives et trop simplistes. Le problème avec le premier choix, est que ces outils ne sont disponibles que pour quelques langues (les plus utilisées). La tâche 2 de l'atelier ACL MultiLing 2013, résumé mono-document multilingue, vient tester la 2ème solution : simpliste. En effet, le corpus intègre un total de 40 langues et il

n'est évidemment pas possible de trouver des outils spécifiques à chaque langue. Pour cette tâche, (Conroy et al., 2013) classent les 40 langues en trois catégories et utilisent leurs outil FASST-E adapté.

Les résultats de cette campagne, résumé mono-document multilingue, étaient plutôt faibles, aucun système n'a pu dépasser la Baseline pour l'anglais par exemple (Kubina, Conroy, & Schlesinger, 2013). Les organisateurs expliquent cette faible performance par le fait que les systèmes proposés jusqu'ici étaient un peu trop adaptés à la structure des articles de presse (pyramide inversée et l'utilisation du critère "position de la phrase").

Dans le présent article, nous proposons un système de résumé multilingue. Notre système utilise un minimum de traitements linguistiques et repose essentiellement sur des traitements numériques pour la pondération et l'extraction de phrases. En effet, le système procède en premier lieu au groupement des phrases en clusters. En deuxième lieu, les termes les plus discriminants vont être les mieux notés afin de mettre en avant les phrases saillantes. On utilise ainsi une méthode d'analyse discriminante: **mRMR** pour **R**edondance **m**inimale et **P**ertinence **M**aximale (Ding, Peng, 2003). Nous proposons également, un nouvel algorithme d'extraction de phrases à deux vitesses : lente et rapide adaptable à la taille du résumé : court ou très court. La prochaine section détaille notre proposition.

3 Description du système

3.1 Pondération des termes

Le but de l'analyse mRMR est la sélection d'un sous ensemble de variables qui représente au mieux l'espace total de variables. A chaque variable sont assignés deux scores: pertinence et redondance. Notre adaptation de cette méthode consiste à donner des scores à chaque terme de façon à sélectionner les termes les plus informatifs. Un terme est d'autant plus informatif si, étant donné les clusters de phrases similaires, sa fréquence varie significativement d'un cluster à un autre et au même temps n'attire pas un grand nombre de termes.

Le choix de l'algorithme de regroupement est primordial pour le succès ou l'échec de mRMR. En effet, la construction de groupes homogènes de phrases conduira naturellement à une meilleure estimation des scores. Ici, nous avons utilisé la classification ascendante hiérarchique et la similarité cosinus entre phrases. Bien entendu, l'utilisation d'un autre algorithme de classification/mesure de similarité est tout aussi possible.

On définit la pertinence d'un terme comme la valeur de l'information mutuelle entre ce terme *t* et la variable de classification *h* [1]. Si le terme et la variable de classification sont fortement corrélés alors le terme *t* est pertinent et nous permet ainsi de décrire au mieux les thèmes évoqués dans le texte.

$$Pertinence\ (t) = I(t,h) \tag{1}$$

Afin de calculer l'information mutuelle entre le terme et la variable de classification, il est nécessaire de construire la matrice des fréquences : M [Phrases ×Termes] où chaque ligne correspond à une phrase et chaque colonne représente un terme. La cellule M[i,j] contient la fréquence d'apparition du $j^{ème}$ terme dans la $i^{ème}$ phrase. A l'issu du groupement, chaque phrase devient rattachée à une classe bien spécifique. Ainsi, chaque ligne de la matrice est augmentée par la valeur de la variable de classification : le numéro de la classe.

Cependant, le fait qu'un terme décrit bien la variation des classes n'est pas suffisant. En effet, il faudra sélectionner des termes pertinents mais les plus dissimilaires possibles. On cherche ainsi à minimiser l'information mutuelle entre un terme donné et le reste des termes. La redondance d'un terme est définie par la moyenne de l'information mutuelle que ce terme partage avec les autres [2]:

$$Redondance\ (t) = \frac{1}{|T-\{t\}|}\sum_{j\in T-\{t\}} I(t,t_j) \tag{2}$$

On cherche ainsi à trier les termes de façon à maximiser la pertinence et minimiser la redondance. (Peng et al., 2005) proposent de maximiser soit la différence MID [3] ou le quotient MIQ [4]:

$$MID \overset{\text{def}}{=} max_{t\in T}[Pertinence\ (t) - Redondance(t)] \tag{3}$$

$$MIQ \overset{\text{def}}{=} max_{t\in T}[Pertinence\ (t)\ /\ Redondance(t)] \tag{4}$$

Le résultat de cette étape est le vecteur des termes avec leurs poids associés $V_{mRMR} = (w_{t1,}\ w_{t2}, \dots\dots\dots\dots, w_{tn})$

3.2 Pondération des phrases

Etant donné le vecteur V_{mRMR}, la tâche est d'attribuer des scores à chaque phrase selon les poids mRMR de ses termes. Pour ce faire, nous calculons la moyenne des poids mRMR des termes de la phrase [5]:

$$Score(P) = \frac{1}{n_p} \sum_{i=1,n_p} w_{t_i}, t_i \epsilon V_p \quad (5)$$

3.3 Extraction des phrases

Dans notre système, le terme est l'unité de calcul des scores et la phrase est l'unité d'extraction. Le résumé sera ainsi formé de n phrases dans la limite de la taille du résumé requise (nombre de caractères, nombre de mots ou pourcentage de compression). Nous définissons deux vitesses d'extraction : rapide et lente. La première, rapide [6], remet à zéro les poids des termes dans V_{mRMR} dés leur intégration au résumé.

$$\forall t \in \{V_{mRMR} \cap V_{p,}\}, w'_t = 0 \quad (6)$$

La deuxième, lente [7], diminue le poids des termes selon le score de la dernière phrase choisie (P). La première vitesse convient dans le cas où on doit générer des résumés très courts car sinon on risque de sélectionner du bruit. La deuxième stratégie sélectionne progressivement les phrases et donne plus de *temps de vie* au terme.

$$\forall t \in \{V_{mRMR} \cap V_{p,}\}, w'_t = w_t - sim(V_{mRMR}, V_{p,}) * w_t \quad (7)$$

3.4 Contexte Multilingue

Dans le contexte multilingue, la dépendance de la langue doit être minimale. En effet, dans le système que nous proposons la dépendance est située au niveau du prétraitement. Notre approche requiert au *minimum* un découpage *correct* en phrases et une analyse lexicale et au *maximum* le filtrage des mots vides et la racinisation. Dans nos expérimentations, nous allons étudier l'impact de l'intégration de ces tâches de manière plus ou moins poussée.

4 Evaluation

Le corpus TAC MultiLing 2011 (Giannakopoulos et al., 2011) contient 10 groupes de documents à résumer. Chaque groupe contient à son tour 10 articles de presse décrivant une séquence de nouvelles autour du même événement: les attaques à la bombe de Londres en 2005 ou le tsunami de 2004, etc. Les textes dans les autres langues ont été traduits de l'anglais par des locuteurs natifs en suivant l'approche phrase par phrase. Trois résumés modèles sont fournis dont la taille est comprise entre 240 et 250 mots.

4.1 Protocole d'évaluation

Notre objectif étant l'évaluation de l'impact de la qualité des traitements linguistiques appliqués sur notre méthode: découpage en phrases naïf ou adapté à la langue source, segmentation en mots ou utilisation d'un tokeniseur, le filtrage des mots vides et enfin l'application ou non de la racinisation. Nous avons pu tester notre approche sur trois langues : l'anglais, l'arabe et le français dans la limite de la disponibilité des outils. Le tableau suivant décrit les différents outils utilisés pour chaque langue :

	Découpage en Phrases	Analyse lexicale	Filtrage des mots vides	Racinisation
Anglais	Stanford tokenizer	Stanford tokenizer	Liste de 571 mots	Porter Stemmer
Arabe	Expression régulière	Expression régulière	Liste de 168 mots	Shereen Khodja Stemmer
Français	Europarl LinguaSentence	Expression régulière	Liste de 127 mots	Snowball Stemmer

TABLE 1 : Outils TAL pour l'anglais, l'arabe et le français

Nous avons donc défini 4 niveaux d'analyse, et pour chaque niveau nous avons généré les résumés en utilisant notre approche:

- **SR**: Découpage en phrases naïf (arabe) ou adapté (anglais, français)
- **TSR**: SR+Analyse lexicale adaptée (anglais, français) ou simple segmentation en mots (arabe)
- **WSR**: TSR+Filtrage des mots vides
- **SSR**: WSR+ racinisation

4.2 Résultats et Discussion

Les tableaux suivants montrent les résultats ROUGE-1 et ROUGE-2 (Lin, 2004) des résumés produits par notre système. Il est à noter que les résumés ont été générés avec un groupement à 8 clusters, vitesse rapide.

	ROUGE-1	ROUGE-2
	F1-score	F1-score
CLASSY	**0,48361**	**0,17612**
BASELINE	**0,40343**	**0,11445**
SR.MRSYN8	0,42056	0,11647
TSR.MRSYN8	0,43531	**0,12739**
WSR.MRSYN8	**0,44111**	0,12512
SSR.MRSYN8	0,4398	0,12671

Anglais

	ROUGE-1	ROUGE-2
	F1-score	F1-score
CLASSY	**0,34296**	**0,14207**
BASELINE	**0,26788**	**0,08982**
SR.MRSYN8	0,2743	0,09492
TSR.MRSYN8	**0,2893**	0,09736
WSR.MRSYN8	0,28125	**0,10044**
SSR.MRSYN8	0,27196	0,08494

Arabe

	ROUGE-1	ROUGE-2
	F1-score	F1-score
JRC	**0,43539**	**0,17199**
BASELINE	**0,37324**	**0,09346**
SR.MRSYN8	0,37324	0,10428
TSR.MRSYN8	0,39482	0,11074
WSR.MRSYN8	**0,39508**	**0,11917**
SSR.MRSYN8	0,3463	0,0935

Français

TABLE 2 : Résultats ROUGE-1 et ROUGE-2 pour l'anglais, l'arabe et le français

On constate que les meilleurs résultats sont obtenus avec le filtrage des mots vides. Ils dépassent la Baseline mais restent en dessous du meilleur système. De plus, la différence entre les 4 variantes n'est pas significative. Nous avons donc évalué le nombre de phrases en commun entre les résumés R1 et R2 issus de deux niveaux consécutifs à l'aide de la formule suivante [8]:

$$Accord(R1, R2) = \frac{2|R1.R2|}{|R1| + |R2|} \quad (8)$$

Le tableau suivant présente les résultats de l'estimation de la moyenne d'accord entre les résumés produits à chaque niveau d'analyse avec le niveau qui le suit:

	Arabe	Anglais	Français	Moyenne
SR×TSR	47,09%	39,78%	39,74%	42,20%
TSR×WSR	38,86%	32,98%	40,91%	37,58%
WSR×SSR	30,78%	44,94%	50,69%	42,13%

TABLE 2 : Variation des phrases extraites avec les niveaux de traitements TAL

On constate qu'avec un seul traitement à la fois, en moyenne 40% des phrases sont maintenues. La plus grande différence (ou désaccord) est enregistrée pour la langue arabe avec l'application du racinisateur. Le plus grand accord est enregistré pour le français avec l'application du racinisateur Snowball. Pour l'anglais, la plus grande différence est enregistrée avec le filtrage des mots vides. Ceci est dû à la taille de la liste des mots vides particulièrement longue pour cette langue. Ainsi, malgré les performances ROUGE très proches des résumés issus de chaque étape (SR, TSR, WSR et SSR), le niveau d'accord est en moyenne 40% et ne dépasse pas un maximum de 50%.

Il est également intéressant de noter que les différents traitements effectués ont un impact non négligeable sur la notation et par conséquent le choix des phrases à extraire. Plus de 70% des phrases choisies au départ (SR) ne le sont pas après traitement (SSR). On remarque aussi qu'avec la racinisation le constat est à l'opposé pour l'arabe et pour le français : on y enregistre la plus grande différence pour l'arabe et le plus grand accord pour le français. En analysant les textes après racinisation, on remarque que le racinisateur Snowball a opéré de faibles modifications morphologiques alors que le racinisateur pour l'arabe extrait la racine sémitique à 3 ou 4 lettres. Ainsi, la qualité des outils utilisés est aussi un paramètre à prendre en compte lors de l'analyse de l'apport des traitements associés.

5 Conclusion

Dans cet article, nous avons évalué l'impact des différents traitements linguistiques de base sur la tache du résumé multi-documents multilingue. Les performances de notre système dépassent la Baseline. Les meilleurs résultats ont été enregistrés avec le filtrage des mots vides, des améliorations sont à prévoir afin de pénaliser les mots vides en utilisant mRMR.

Malgré les scores ROUGE pratiquement équivalents, le choix des phrases pour la construction des résumés varie considérablement d'une étape à une autre et ceci pour les trois langues. Cette variation s'explique par : la variation du score des formes que prend chaque terme à travers les différents niveaux d'analyse d'une part et par la qualité variable des outils utilisés pour chaque traitement d'une autre part. De plus, la variation des performances du système d'une langue à une autre est à étudier, les résultats pour l'arabe sont 10% moins bons que pour les autres langues, est ce que c'est uniquement du à la qualité des outils utilisés pour chaque langue ? Doit-on considérer la stabilité des performances à travers les langues comme critère d'évaluation ?

L'évaluation de l'efficacité du système et de la qualité des résumés produits à chaque étape nécessite une évaluation manuelle, les scores ROUGE étant pratiquement équivalents malgré la grande variation des choix de phrases fait resurgir de nouvelles questions. En effet, on constate de plus en plus que procéder à une évaluation automatique en utilisant ROUGE n'est pas suffisant. Elle doit être complétée par une évaluation manuelle pour juger des critères autre que la couverture lexicale : qualité linguistique, cohérence, etc.

Références

BARZILAY R., MCKEOWN K. R. (2005). Sentence Fusion for Multidocument News Summarization. *Comput. Linguist.*, *31*(3), 297–328.

BOUDIN F., TORRES-MORENO J.-M. (2009). Résumé automatique multi-document et indépendance de la langue: une première évaluation en français. Actes de *Traitement Automatique de La Langue Naturelle (TALN'09)*

CONROY J., DAVIS S. T., KUBINA J., LIU Y.-K., O'LEARY D. P., SCHLESINGER J. D. (2013). Multilingual Summarization: Dimensionality Reduction and a Step Towards Optimal Term Coverage. Actes de *MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, 55–63.

CONROY J. M., SCHLESINGER J. D., KUBINA J., RANKEL P. A., O'LEARY, D. P. (2011). CLASSY 2011 at TAC: Guided and multi-lingual summaries and evaluation metrics. Actes de *Text Analysis Conference*.

DAS P., SRIHARI R. (2011). Global and Local Models for Multi-Document Summarization. Actes de *Text Analysis Conference*.

DING C., PENG H. (2003). Minimum Redundancy Feature Selection from Microarray Gene Expression Data. Actes de *IEEE Computer Society Conference on Bioinformatics (CSB'03)*, 523–529.

GIANNAKOPOULOS G., EL-HAJ M., FAVR, B., LITVAK M., STEINBERGER J., VARMA V. (2011). TAC 2011 MultiLing Pilot Overview. Actes de *Text Analysis Conference (TAC2011)*.

HMIDA F., FAVRE B. (2011). LIF at TAC Multiling: Towards a Truly Language Independent Summarizer. Actes de *Text Analysis Conference (TAC)*.

LEDENEVA, Y., 2008. Effect of preprocessing on extractive summarization with maximal frequent sequences, *Actes de MICAI 2008: Advances in Artificial Intelligence*, 123–132.

LIN, C. (2004). Rouge: A Package for Automatic Evaluation of Summaries. Actes de *Text Summarization Branches Out: ACL-04 Workshop*, 74–81.

LITVAK M., LAST M., FRIEDMAN M. (2010). A New Approach to Improving Multilingual Summarization Using a Genetic Algorithm. Actes de *48th Annual Meeting of the Association for Computational Linguistics*, 927–936.

MARCU D. C. (1998). The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. Actes de *35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 96-103.

MIHALCEA R., TARAU P. (2005). A Language Independent Algorithm for Single and Multiple Document Summarization. Actes de *International Joint Conference on Natural Language Processing (IJCNLP)*, Vol. 5.

PENG H., LONG F., DING C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*(8), 1226–1238.

SAGGION H. (2011). Using SUMMA for Language Independent Summarization at TAC 2011. Actes de *Text Analysis Conference.*

Porting a Summarizer to the French Language

RÉMI BOIS[1] JOHANNES LEVELING[2] LORRAINE GOEURIOT[2] GARETH J. F. JONES[2]
LIADH KELLY[2]
(1) LINA, Université de Nantes, France *
(2) CNGL, School of Computing, Dublin City University, Dublin 9, Ireland
remi.bois@etu.univ-nantes.fr, {lgoeuriot, lkelly, gjones, jleveling}@computing.dcu.ie

Résumé. Nous présentons dans cet article l'adaptation de l'outil de résumé automatique REZIME à la langue française. REZIME est un outil de résumé automatique mono-document destiné au domaine médical et s'appuyant sur des critères statistiques, syntaxiques et lexicaux pour extraire les phrases les plus pertinentes. Nous décrivons dans cet article le système REZIME tel qu'il a été conçu et les différentes étapes de son adaptation à la langue française. Les performances de l'outil adapté au français sont mesurées et comparées à celle de la version anglaise. Les résultats montrent que l'adaptation au français ne dégrade pas les performances de REZIME, qui donne des résultats équivalents dans les deux langues.

Abstract. We describe the porting of the English language REZIME text summarizer to the French language. REZIME is a single-document summarizer particularly focused on summarization of medical documents. Summaries are created by extracting key sentences from the original document. The sentence selection employs machine learning techniques, using statistical, syntactic and lexical features which are computed based on specialized language resources. The REZIME system was initially developed for English documents. In this paper we present the summarizer architecture, and describe the steps required to adapt it to the French language. The summarizer performance is evaluated for English and French datasets. Results show that the adaptation to French results in system performance comparable to the initial English system.

Mots-clés : Résumé automatique, multilangue, domaine médical.

Keywords: single-document summarization, multilingual, medical domain.

1 Introduction

The rapid growth of online text resources is producing information overload, where individuals cannot make use of all the available information. This is particularly the case in specialised domains such as medicine and biomedicine, where finding relevant information is critical. As stated by Afantenos *et al.* (2005), "the number of scientific journals in the fields of health and biomedicine is unmanageably large, even for a single speciality". This makes it very difficult for scientists to follow the evolution of their domain. Laypersons seeking medical information from the internet are facing a similar situation. Automatic summarization for the medical domain is a possible mechanism towards alleviating this problem.

In this paper we describe the French language version of the system REZIME, an automatic summarizer designed to create efficient summaries of documents from the medical domain. It generates single-document summaries, built from sentences containing key material extracted from documents. Sentence selection is based on machine learning techniques, using statistical, syntactic and lexical features computed with specialized language resources. REZIME was initially developed for English documents (Nguyen & Leveling, 2013). While its architecture is language-independent, porting it to another language requires adaptation and/or translation of its linguistic resources (i.e. syntactic, lexical and terminological resources). In this paper we present the different steps required to adapt it to another language and with a specific focus on adapting the summarizer to the French language, and show its comparable effectiveness with the original English language system.

The remainder of this paper is organized as follows : After a brief description of the main studies in single-document summarization in Section 2, we describe the REZIME architecture in Section 3. The main steps involved in its adaptation to French are presented in Section 4. The evaluation of the French summarizer and the comparison of its performance to the English version are shown in Section 5. We conclude with remarks on future work in Section 6.

∗. This study has been conducted while Rémi Bois was an intern in CNGL.

2 Related work

Automatic summarization is defined as the process of creating "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that" (Radev *et al.*, 2002). The two main approaches to text summarization are extractive summarization and abstractive summarization. Extractive summarization consists in selecting parts of the original text that contain the most important information which then form the summary. Extractive methods rely mostly on machine learning, using a set of features to rank sentences (Mani & Bloedorn, 1998). Abstractive summarization aims at rephrasing the content of the texts, generating sentences that do not necessarily appear in the original document (Barzilay & Mckeown, 2005).

While much summarization research has focused on the English language, the proliferation of online content in other languages is creating a growing demand for multilingual summarization system. This topic has been explored in various work, mostly concentrating single document summarisation of news corpora (Dalianis *et al.*, 2004; Litvak *et al.*, 2010). Existing work in French summarization has mostly aimed at providing multilingual or cross-lingual summaries (Torres-Moreno *et al.*, 2001; Fernandez *et al.*, 2008; Boudin & Torres-Moreno, 2009).

Summarization in specialised domains brings specific issues, with medicine raising unique challenges. As stated by Afantenos *et al.* (2005) "uniqueness of medical documents is due to their volume, their heterogeneity, as well as due to the fact that they are the most rewarding documents to analyse, especially those concerning human medical information due to the expected social benefits". While some systems for medical summarization already offer multilingual summaries, for example using English documents to create French summaries (Lenci *et al.*, 2002), our work focuses on the language specific porting of an English language summarizer tuned to the medical domain.

3 A single document summarizer for medical text

REZIME is a single-document extractive summarizer that was initially designed for the English language, with specific features developed for the medical domain (Nguyen & Leveling, 2013). The medical summaries generated by REZIME are presented to patients and medical professionals. Therefore, the readability and clarity of the summaries is critical. In the design of REZIME, we opted for an extractive approach since non-extractive ones may lead to incoherent sentences, potentially giving inaccurate information to the user if it is incorrectly extracted from the original document.

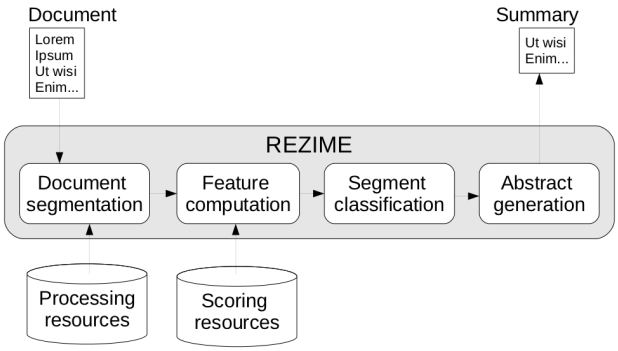


FIGURE 1: REZIME Workflow

For summarization, each document is processed in a workflow consisting of four steps, illustrated in Figure 1. First, the document is preprocessed to extract its structure (paragraphs, sentences, tokens). Each sentence is then represented by a vector of features, that can either be statistical or linguistic.

Seven term checking features are used to detect the presence or absence of particular significant words or phrases in sentences. It checks for the presence of pre-defined basic words that can help to create different summaries for a general audience or for professionals, and searches for cue phrases such as "importantly" or "in summary" as they are good indicators of whether a sentence should be included in a summary. It also counts the overlap of a sentence with the title terms as a feature. A naive Named Entity recognition tool checks for the number of capitalized words (the first word of the sentence is excluded). This naive approach is used since the summarizer is designed to work online and process summaries on the fly, so speed and robustness are significant issues. Preposition detection is used since they often give context to sentences. Finally, pronoun detection and counting of punctuation marks are used to discard sentences that contain too many of these features, which can make them hard to interpret independently. Most of these features are commonly used in summarization for newspaper articles (Lin, 1999).

Five non term checking features are used : a keyword cluster feature based on a method proposed by (Luhn, 1958) to score sentences according to the number of significant words they contain ; the global bushy feature, which generates inter-document links based on similarity of paragraphs (Salton *et al.*, 1997) ; count of the number of terms in each sentence, assuming that sentences which are too long or too short are less useful ; the position of a sentence in a paragraph, since sentences occurring early in a paragraph usually give context for all the paragraph ; and finally, TF-ISF, which is a sentence-level equivalent to TF-IDF.

Two features specific to the medical domain are also included : the affix presence feature, checking for terms containing medically-related affixes (896 affixes) ; and the domain term feature, checking for terms appearing in a list of medical terms (5,799 terms).

These features are then aggregated through a machine learning algorithm, in which the selection factors are combined in a weighted linear summation. Since development of the best machine learning algorithm is still in progress, we set all weights to 1 for this work. Finally, the selected sentences are post processed to provide a readable summary.

Some of these linguistic features rely on language-dependent resources. Their adaptation to the French language, is described in the following section.

4 Porting REZIME to the French Language

In this section, we describe the resources used by REZIME and their adaptation to the French language. Summarizers architectures and features are often language-independent, e.g. keyword or phrase matching, keyword clustering or the global bushy algorithm. The major challenges of porting a summarizer to a new language relate mainly the availability of the necessary linguistic resources in this language. Development of these resources can generally be achieved by different methods : (i) using an existing resource in the target language ; (ii) manually translating an existing resource ; (iii) automatically translating an existing resource ; or (iv) creating a new resource.

Option (iii) is in most cases not viable, especially for domain-specific resources in languages other than English. Option (iv) is costly, both in terms of time and manual labour. Thus, for each resource, we opted for the first two methods.

4.1 Resources used in the REZIME System

Category	Resource name	Use
Preprocessing resources	poss_sent_end	Indicates which tokens can end a sentence
	bad_sent_start	Indicates an impossible beginning for a sentence
	bad_sent_end	Indicates an impossible end for a sentence
	abbreviations	List of common abbreviations
Feature computation resources	sections	Potential title of sections in a paper
	cue_phrases	Keywords indicating that a sentence may be important
	medical_dict	Affixes indicating a medical term (eg patho-, -trophy)
	medical_terms	List of medical terms
	SpacheWordList	List of most common words
	stopwords	List of stopwords

TABLE 1: Resources for the Summarizer

The resources used by the summarizer can be divided into two categories : document processing resources and scoring resources. Table 1 gives a list of the specific resources used and the category they belong to. The first category includes keywords used for sentence boundary detection. These indicate how and where to split a text into paragraphs, a paragraph into sentences, and a sentence into tokens. Most of these resources include generic vocabulary and/or punctuation marks. The *poss_sent_end* resource contains punctuation marks that can appear at the end of a sentence (e.g.. '.', ' ?'). The abbreviation resource is composed of frequent acronyms and abbreviations (e.g.. 'Mr.', 'Dr.'). Most of the resources from this category are language independent (e.g. punctuation marks), at least among European languages.

The second category contains the resources used to represent sentences as vectors of features. These indicate sentence content that should be included in a summary. The *cue_phrases* resource contains a list of key phrases such as 'in conclusion', 'the most important'. The *SpacheWordList* resource lists frequent and general terms that should be preferred in a summary intended for non-specialists, e.g.. 'after', 'again'. Two scoring resources are specialized for the medical domain. Affixes for medical terms, which allow recognition of scientific terms, are created by using Greek and Latin elements (Andrews,

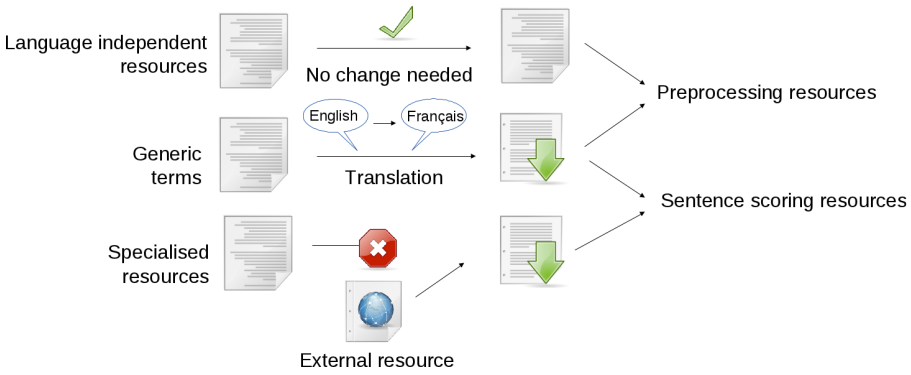


FIGURE 2: Strategies to port resources to the French language

1948). Nearly 900 of these affixes are gathered in our *medicat_dict* resource (e.g.. 'patho-', '-trophy'). The *medical_terms* list consists of several thousands of medical terms, drug names and symptoms, collected from `www.medterms.com`.

These resources are mainly language dependent. Therefore they needed to be adapted to the French language. As described in Figure 2, this requires a terminological and syntactical adaptation (translation), or sometimes involves including new resources. We describe this process in the following sections.

4.2 Porting Resources to the French language

4.2.1 Preprocessing Resources

Preprocessing resources include elements enabling REZIME to split paragraphs into sentences and sentences into word tokens, shown in Table 1. These resources are not specialized and need simple translation from English. Some of these tools did not need any modification since they contain only punctuation (*poss_sent_end*, *bad_sent_start* and *bad_sent_end*). The only resource translated is the list of general abbreviations and acronyms (*abbreviations*), which were manually translated by a native French speaker.

4.2.2 Adaptation of feature extraction resources

Feature extraction include elements used to represent sentences as feature vectors, in order to choose which ones should be included in a summary, as shown in Figure 1. These language specific resources cannot be automatically translated with a high enough accuracy for use in the REZIME system due to the presence of specialized vocabulary or specific linguistic features, e.g. medical terms, cue phrases. As shown in Figure 2, resources can either be (i) kept as they are ; (ii) translated ; or (iii) replaced by a French resource.

1. The list of medical affixes *medical_dict* did not have to be translated. These are Latin and Greek affixes (e.g. cephal-, coron-, -phage), that are also widely used to create French medical terms.
2. We manually translated *sections* and *cue_phrases*. These resources were short lists and a manual translation was the most cost-effective solution (e.g. significant).
3. Since similar resources were available in French, we replaced *medical_terms*, *stopwords* and *SpacheWordList*. The medical_terms resource was created in a European project [1]. It contains 1,840 medical terms, available in 8 languages, including French and English. The stopwords list is a short list of 126 terms. The SpacheWordList is composed of 1,750 words from the general language [2].

5 Evaluation and Comparison of the English and the French Summarizers

The goal of this experiment is to assess the effectiveness the ported REZIME summarizer, by investigating its performance relative to the original English summarizer. While an evaluation on a parallel test collection in two languages would have been ideal, we did not have access to such a collection. We test our system on two independent collections, both from the medical domain, but on two different specialities (the English one is generic, the French one deals with endoscopy).

1. `http://users.ugent.be/~rvdstich/eugloss/welcome.html`
2. `http://fr.wiktionary.org/wiki/Wiktionnaire:Liste_de_1750_mots_fran\%C3\%A7ais_les_plus_courants`

	French	English
Origin	Acta Endoscopica	BioMed central
Corpus size (documents)	104	100
Average length of Manual Abstracts (words)	220	250
Average length of REZIME Summaries (words)	218	230

TABLE 2: Statistics of the French and English corpora and summaries.

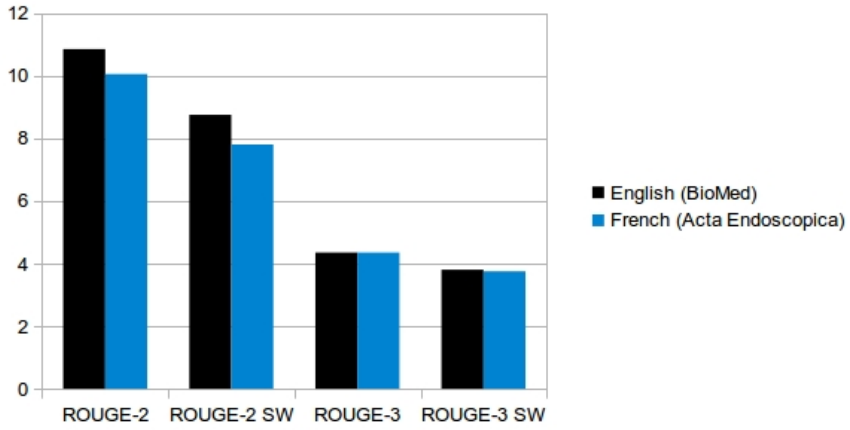


FIGURE 3: ROUGE scores for the English and the French summaries with REZIME

Our assumption is that the topical focus will not affect the results significantly, since the resources come from the same domain. In this section, we describe the two corpora used for our experiments and the encouraging results we obtained.

5.1 Evaluation Test Collections

To evaluate the similarity of the two systems, we used comparable corpora from the medical domain. We created a corpus for the French language from scientific articles on a medical speciality journal : Acta Endoscopica. The English corpus was composed of articles from BioMed Central (BMC) [3]. We assume a manually written abstract is a good quality summary of the scientific articles, and use these as a gold standard reference for automatic summarization to evaluate our system (da Cunha & Wanner, 2005).

The automatically created summaries are limited to a length of about 230 words which is comparable in length to the manual abstract provided with the documents. Our test sets consists of 100 documents in each language. Table 2 provides some statistics on the corpora.

5.2 Results

We measure the quality of the automatically created summaries using the ROUGE measure (Lin, 2004). We use ROUGE-2 and ROUGE-3 with and without considering stopwords, as they are the most widely used ROUGE scores. We are interested in changes that porting to French causes to the behaviour of the REZIME system. Figure 3 shows the experimental results.

We observe that the ROUGE-3 scores are almost identical for the two systems. The ROUGE-2 score drops by one point at most for the French system, we believe this is mostly due to the differences between corpora, the French corpora belonging to a more specific medical area, with terms that do not appear in our list of medical terms. ROUGE-3 gives lower results than ROUGE-2, as would be expected, since it is based on 3-grams instead of bigrams.

This experiment shows that porting the summarizer to another language leads to similar summarization performance. The lack of parallel data and the consequent use of varied evaluation collections could affect the results, but additional experiments would be required to investigate this potential effect.

3. http://www.biomedcentral.com

6 Conclusion and Future Work

In this paper, we described methods to adapt an extractive summarizer to another language. We presented results showing that single document summarization systems based on extraction can be successfully ported to an alternative language. Our adaptation technique is based on selecting and generating adequate resources by translation, which is an adequate approach even when the system deals with a specific context like the medical domain. We also created a new summarization evaluation collection, consisting of French documents from the medical domain.

As part of future work, we plan to investigate whether the technique which has been shown to be successful for French is also applicable to other pairs of languages, and to other domains.

Acknowledgments

This work is supported in part by the Khresmoi project (257528) and SFI (07/CE/I1142) as part of the Centre for Next Generation Localisation at Dublin City University.

Références

AFANTENOS S., KARKALETSIS V. & STAMATOPOULOS P. (2005). Summarization from medical documents : a survey. *Artificial intelligence in medicine*, **33**(2), 157–177.

ANDREWS E. (1948). A history of scientific English. *The American Journal of the Medical Sciences*, **215**(1), 120.

BARZILAY R. & MCKEOWN K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, **31**, 297–328.

BOUDIN F. & TORRES-MORENO J.-M. (2009). Résumé automatique multi-document et indépendance de la langue : une première évaluation en français. In *TALN 2009 – Session posters*.

DA CUNHA I. & WANNER L. (2005). Towards the automatic summarization of medical articles in Spanish : Intergration of textual, lexical, discursive and syntactic criteria. *Crossing Barriers in Text Summarization Research, RANLP, Borovets*.

DALIANIS H., HASSEL M., STOCKHOLM K., SMEDT K. D., LISETH A., COGNIT T. C. L., WEDEKIND J. & COPENHAGEN C. (2004). Porting and evaluation of automatic summarization. In *Nordisk Sprogteknologi*, p. 107–121.

FERNANDEZ S., SANJUAN E. & TORRES-MORENO J. M. (2008). Enertex : un système basé sur l'énergie textuelle. In *TALN 2008*, p. 99–108.

LENCI A., BARTOLINI R., CALZOLARI N., AGUA A., BUSEMANN S., CARTIER E., CHEVREAU K. & COCH J. (2002). Multilingual summarization by integating linguistic resources in the MLIS-MUSI project. In *LREC*, volume 2, p. 1464–1471.

LIN C.-Y. (1999). Training a selection function for extraction. In *CIKM 1999*, p. 55–62 : ACM.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In *Text Summarization Branches Out : Proceedings of the ACL-04 Workshop*, p. 74–81.

LITVAK M., LAST M. & FRIEDMAN M. (2010). A new approach to improving multilingual summarization using a genetic algorithm. In *ACL 2010*, p. 927–936.

LUHN H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, **2**(2), 159–165.

MANI I. & BLOEDORN E. (1998). Machine learning of generic and user-focused summarization. In *AAAI/IAAI*, p. 821–826.

NGUYEN D. & LEVELING J. (2013). Exploring domain-sensitive features for extractive summarization in the medical domain. In *Natural Language Processing and Information Systems*, volume 7934 of *LNCS*, p. 90–101.

RADEV D. R., HOVY E. H. & MCKEOWN K. (2002). Introduction to the special issue on summarization. *Computational Linguistics*, **28**, 399–408.

SALTON G., SINGHAL A., MITRA M. & BUCKLEY C. (1997). Automatic text structuring and summarization. *Information Processing & Management*, **33**(2), 193–207.

TORRES-MORENO J.-M., VELÁZQUEZ-MORALES P. & MEUNIER J.-G. (2001). Cortex : un algorithme pour la condensation automatique de textes. *ARCo 2001*, **2**.

Extraction de données orales multi-annotées

Brigitte Bigi[1] Tatsuya Watanabe[1,2]
(1) Laboratoire Parole et Langage, AMU, CNRS, 5 avenue Pasteur, 13100 Aix-en-Provence
(2) Ortolang
brigitte.bigi@lpl-aix.fr, tatsuya.watanabe@atilf.fr

Résumé. Cet article aborde le problème de l'extraction de données orales multi-annotées : nous proposons une solution intermédiaire, entre d'une part les systèmes de requêtages très évolués mais qui nécessitent des données structurées, d'autre part les données (multi-)annotées des utilisateurs qui sont hétérogènes. Notre proposition s'appuie sur 2 fonctions principales : une fonction booléenne pour filtrer sur le contenu, et une fonction de relation qui implémente l'algèbre de Allen. Le principal avantage de cette approche réside dans sa généricité : le fonctionnement sera identique que les annotations proviennent de Praat, Transcriber, Elan ou tout autre logiciel d'annotation. De plus, deux niveaux d'utilisation ont été développés : une interface graphique qui ne nécessite aucune compétence ou connaissance spécifique de la part de l'utilisateur, et un interrogation par scripts en langage Python. L'approche a été implémentée dans le logiciel SPPAS, distribué sous licence GPL.

Abstract. This paper addresses the problem of extracting multimodal annotated data in the linguistic field ranging from general linguistic to domain specific information. Our proposal can be considered as a solution or a least an intermediary solution that can link together requesting systems and expert data from various annotation tools. The system is partly based on the Allen algebra and consists in creating filters based on two functions : a boolean function and a relation function. The main advantage of this approach lies in its genericity : it will work identically with annotations from Praat, Transcriber, Elan or from any other annotation software. Furthermore, two levels of usage have been developed : a graphical user interface graph that not requires any skill or knowledge, and a query form in Python. This system is included in the software SPPAS and is distributed under the terms of the GPL license.

Mots-clés : multimodalité, corpus, extraction.

Keywords: multimodality, corpus, extraction.

1 Introduction

Au cours de ces dix dernières années, la quantité de données linguistiques qui sont devenues disponibles a considérablement augmenté, non seulement pour les corpus écrits, mais aussi pour les données orales. Dans divers domaines de la linguistique, il est aujourd'hui de plus en plus attendu pour les linguistes que leurs études portent sur des quantités importantes de données empiriques, pouvant comprendre jusqu'à plusieurs heures de parole. Jusqu'à il y a quelques années, il était difficile, voire impossible, de manipuler de telles quantités de données. De nombreux logiciels, distribués librement, permettent désormais d'annoter manuellement des corpus oraux, tels que *Transcriber* (Barras *et al.*, 1998), *Praat* (Boersma, 2001), ou *WaveSurfer* (Sjölander et Beskow, 2000), pour ne citer que quelques uns des plus populaires (voir (Llisterri, 2011) pour une étude complète des logiciels d'analyse de la parole). D'autres outils sont plus orientés vers l'annotation manuelle à partir de vidéos, tels que *Elan* (Brugman *et al.*, 2004) ou *Anvil* (Kipp, 2013), mais on ne peut pas les utiliser pour effectuer des annotations phonétiques ou prosodiques précises. D'autres outils, ou boîtes à outils, permettent d'annoter automatiquement de très grandes quantités de données orales, pour divers niveaux d'annotations. Le LPL, par exemple, distribue le logiciel *SPPAS* (Bigi et Hirst, 2012), sous licence GPL, qui permet d'obtenir une segmentation automatique en mots, syllabes et phonèmes, à partir d'enregistrements audio transcrits. Un analyseur morpho-syntaxique y est également développé (Rauzy et Blache, 2009) et permet d'obtenir automatiquement un étiquetage de données orales alignées sur le signal.

La plupart du temps, les données annotées sont représentées sous la forme de "tiers". Ces "tiers" sont composées de séries

d'intervalles (ou de points pour le logiciel Praat). Les intervalles sont définis par un temps de début, un temps de fin et un label. Dans Praat, aucun lien ne peut être défini entre les "tiers". Dans ANVIL, par contre, avant toute utilisation, un schéma d'annotation doit être défini. De cette manière, des dépendances peuvent être établies entre les "tiers".

Ainsi, les annotations depuis ces dernières années ont ceci de particulier qu'elles concernent de plus en plus souvent différents domaines, notamment grâce aux développements logiciels récents. C'est ce qui est constaté dans (O'Halloran et Smith, 2012) : "in recent decades the rapid increase in sophistication and availability of technological (particularly computational) resources and techniques for analysis of multimodal text has no doubt driven the rapid increase in multimodal analyses appearing within a range of disciplines, vastly improving, as technology did for the study of speech earlier, our access to and understanding of multimodal text using, for example, multimodal annotation software".

Les annotations multimodales comprennent des informations qui peuvent concerner divers domaines tels que la prosodie, la phonétique, la syntaxe, le discours, la gestuelle, etc, comme par exemple dans (Blache *et al.*, 2010). Ainsi, le plus grand obstacle aujourd'hui auquel les linguistes doivent faire face n'est pas le stockage de données, ni leur annotation, mais plutôt leur *exploration*. Ce challenge est notamment mentionné dans (O'Halloran, 2011), "The major challenge to Multimodal Discourse Analysis is managing the detail and complexity involved in annotating, analysing, searching and retrieving multimodal semantics patterns within and across complex multimodal phenomena.".

Parallèlement, les moyens pour requêter des données se sont considérablement améliorés, passant de simples bases de données sous formes de tables interrogées à l'aide de langages comme SQL, à des systèmes complexes de stockage de l'information (OWL par exemple) permettant des requêtes élaborées à l'aide de langages (sparql ou Xquery par exemple). Cependant, à notre connaissance, aucun de ces systèmes ne permet l'exploration et l'extraction directement à partir des données issues des logiciels d'annotation.

Dans cet article, nous proposons une solution simple et originale qui permet d'explorer/extraire directement les données multi-annotées des utilisateurs. Le système proposé se situe à mi-distance entre d'une part les systèmes d'interrogations qui ne peuvent opérer que sur des données structurées et les recherches "élémentaires" à base de patrons que l'on trouve dans les différents logiciels d'annotations. Notre système consiste à créer des filtres, puis à les appliquer sur les données annotées. En sortie, il produit une nouvelle annotation en fonction de la sélection effectuée. La version actuelle permet de traiter des données provenant des logiciels d'annotation Transcriber, Praat, Elan ou à partir de fichiers CSV. De plus, d'autres formats pourront s'ajouter à cette liste sans remettre en cause le fonctionnement interne du système.

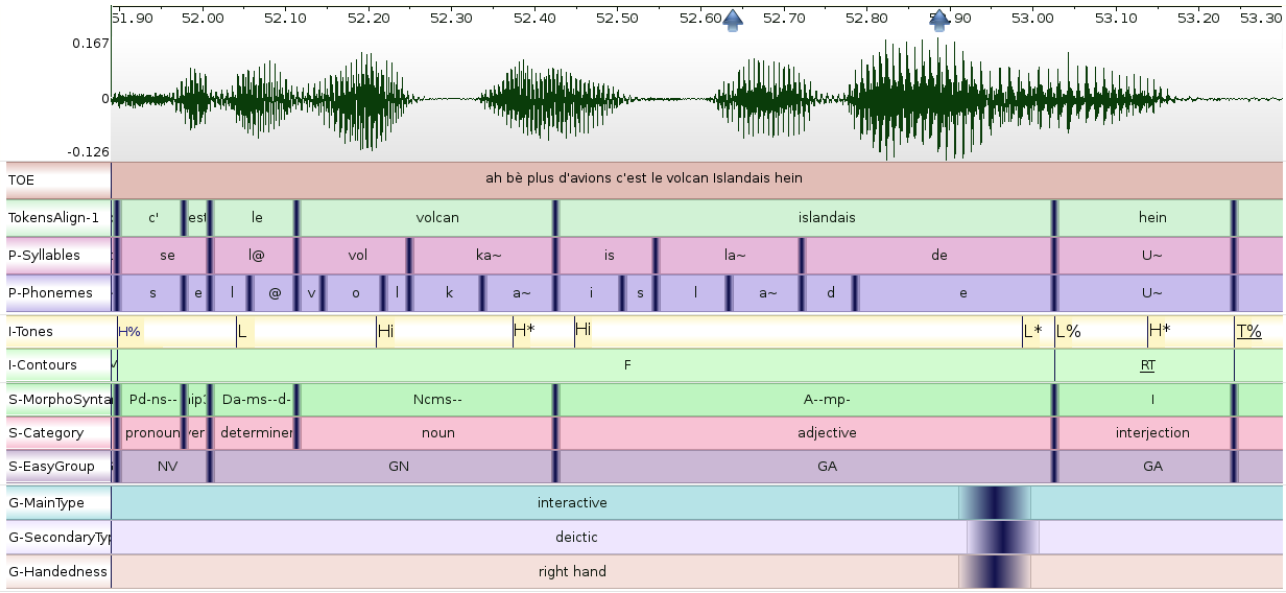


FIGURE 1 – Exemple de données multi-annotées manuellement et automatiquement, importées de 3 logiciels différents.

2 Représentation des données multi-annotées

La difficulté majeure dans le traitement de ces données provient essentiellement de la nature même des annotations : selon le type d'annotations, les bornes des intervalles sont placées de façon plus ou moins précise. Par exemple, lorsqu'il s'agit d'annotations phonétiques ou prosodiques la précision est de l'ordre de quelques millisecondes. D'autres annotations, comme celle des types de gestes, ne peuvent pas être plus précises que la durée d'une image, soit 40 ms la plupart du temps. Pour surmonter cette difficulté, une solution pour représenter les données a été proposée dans (Bigi *et al.*, 2014). Elle consiste à représenter un point dans le temps X à l'aide de deux valeurs : un "midpoint" M_x (valeur centrale) et un "radius" R_x (zone d'incertitude), tel que $X = (M_x, R_x)$.

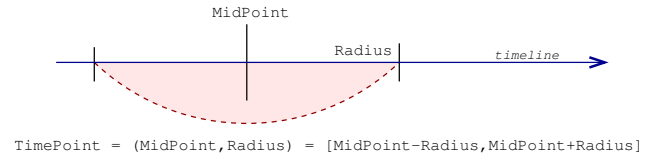


FIGURE 2 – Représentation du temps des données annotées

L'utilisation la plus simple de ces valeurs consiste à attribuer au radius une valeur égale à la moitié de la valeur du taux d'échantillonnage du média annoté (par exemple, 20 ms dans le cas d'une vidéo), ou à la durée que représente 1 pixel à l'écran au moment de l'annotation. Les comparaisons de deux points dans le temps X et Y se feront comme décrit dans l'équation 1.

$$\begin{aligned} X = Y &\Leftrightarrow |M_X - M_Y| \leq R_X + R_Y \\ X < Y &\Leftrightarrow \neg(X = Y) \wedge (M_X < M_Y) \\ X > Y &\Leftrightarrow \neg(X = Y) \wedge (M_X > M_Y) \end{aligned} \quad (1)$$

La figure 1 montre différentes annotations dans lesquelles les bornes utilisent cette notion. La valeur de radius a été fixée à 30 ms pour les annotations gestuelles, 0 ms pour les annotations prosodiques, et 5 ms pour les autres annotations.

Le système d'interrogation proposé dans cet article s'appuie sur cette représentation des données afin de rendre l'extraction plus robuste (en particulier lorsqu'il sera question de faire des extractions multi-domaines).

3 Filtrage sur un seul niveau d'annotation : fonction booléenne

Cette section décrit l'ensemble des filtres qui ont été prévus pour extraire les données d'une seule "tier" (une ligne d'annotation). Nous montrons la syntaxe que nous proposons d'utiliser avec **l'interpréteur Python**. Chacune des fonctionnalités peut être utilisée via **l'interface graphique** (figure 3). Dans la suite, nous noterons $Bool$ une fonction booléenne qui s'applique sur chacune des annotations d'une tier T, à l'aide d'un filtre f.

La recherche à base de patrons constitue une partie importante de tout système d'extraction. Ainsi, les filtres qui suivent sont proposés afin de sélectionner les annotations selon leur label (on note P, le patron à chercher) :

- correspondance exacte : $f = T(Bool(exact = P))$,
- contient : $f = T(Bool(contains = P))$,
- commence par, $f = T(Bool(startswith = P))$,
- termine par, $f = T(Bool(endswith = P))$,
- expression régulière, $f = T(Bool(regexp = R))$.

Les fonctions booléennes peuvent être inversées (si elles sont précédées par le caractère $\sim$). De plus, les recherches peuvent être sensibles à la casse ou non (sauf le dernier), si le critère commence par 'i', par exemple : $f = T(\sim Bool(iexact = P))$.

Deux types de filtres peuvent être créés avec des critères de temps : en fonction de la durée d'une annotation, ou de ses valeurs de début et de fin :

- une durée inférieure à une valeur v : $f = T(Bool(duration\_lt = v))$,
- une durée inférieure ou égale à une valeur v :$f = T(Bool(duration\_le = v))$,
- une durée supérieure à une valeur v : $f = T(Bool(duration\_gt = v))$,

— une durée supérieure ou égale à une valeur v : $f = T(Bool(duration\_ge = v))$,
— une durée égale à une valeur v : $f = T(Bool(duration\_e = v))$,
— le début doit être après ou égale à un temps t : $f = T(Bool(begin\_ge = t))$,
— la fin doit être avant ou égale à un temps t : $f = T(Bool(end\_le = t))$.

L'implémentation des opérateurs & ("ET") et | ("OU") permet de combiner ces critères. Par exemple, on pourra écrire : $f = T(((Bool(exact = P_1) \mid Bool(exact = P_2))) \ \& \ Bool(duration\_lt = v))$.

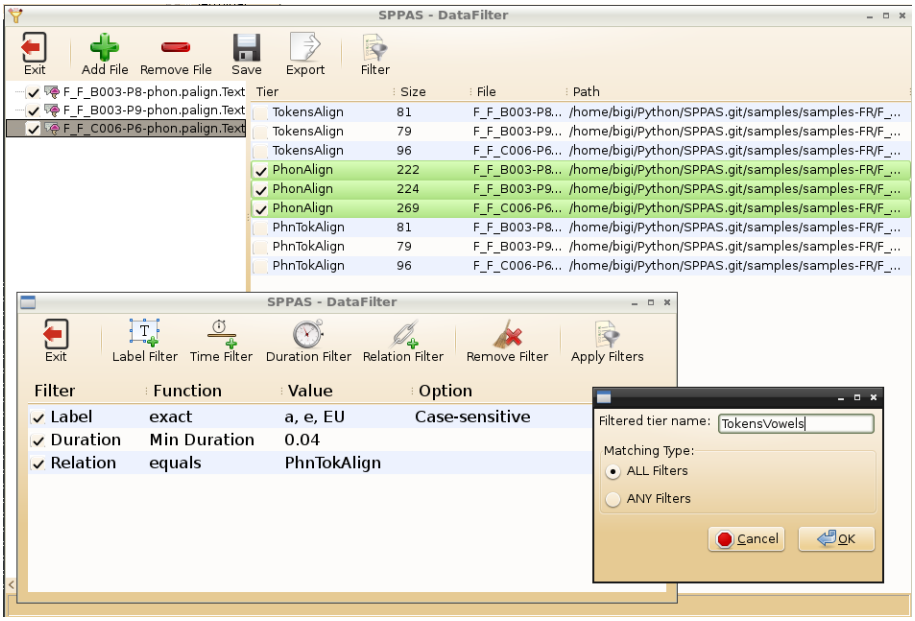


FIGURE 3 – Filtrage à l'aide de l'interface graphique.

4 Filtrage entre différents niveaux d'annotations : fonction relation

Dans la théorie de Allen, les manières d'ordonner les extrémités de deux intervalles sont décrites par 13 relations atomiques (before, after, meets, met by, overlaps, overlapped by, starts, started by, finishes, finished by, contains, during, equals). L'implémentation que nous proposons des relations de Allen est décrite dans la table 1, où l'on note $X = [X^-, X^+]$ et $Y = [Y^-, Y^+]$ deux intervalles non réduits à un point. Nous avons étendue notre implémentation aux 25 relations du modèle INDU ("INterval and DUration"), proposé dans (Pujari *et al.*, 2000), qui intègre des contraintes de durées des intervalles aux relations de Allen : X est de durée inférieure, supérieure ou égale à Y.

De plus, nous avons ajouté des critères de délai pour les relations "after" et "before", ainsi que pour les relations "overlaps" et "overlapped by".

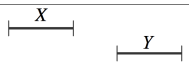
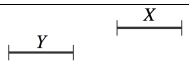
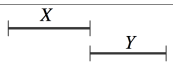
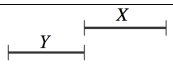
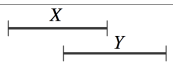
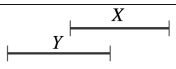
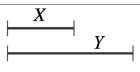
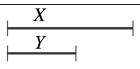
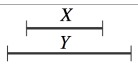
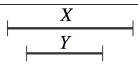
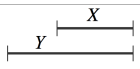
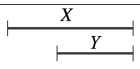
Pour simplifier l'utilisation, nous proposons 3 meta-règles : "equals", "convergent" et "disjoint". La meta-relation convergent inclue overlaps, overlapped by, starts, started by, during, contains, finishes et finished by. La meta-relation "disjoint" inclue before, after, meets et met by.

Enfin, dans la mesure où certains logiciels proposent d'utiliser des annotations sous forme de points plutôt que d'intervalles, nous avons implémenté cette possibilité.

5 Exemple de requête

À titre d'illustration, cette section présente l'implémentation Python d'une requête, sachant que celle-ci peut tout aussi bien être réalisée avec l'interface graphique : **Quels sont les gestes pendant les pauses ?**

TABLE 1 – Relations de Allen entre deux intervalles, et leur implémentation à l'aide de deux filtres f_1 et f_2

Filtre relationnel	Illustration	Description
$f_1.Link(f_2, Rel("before"))$		$X^+ < Y^-)$
$f_1.Link(f_2, Rel("after"))$		$(X^- > Y^+)$
$f_1.Link(f_2, Rel("meets"))$		$(X^+ = Y^-)$
$f_1.Link(f_2, Rel("metby"))$		$(X^- = Y^+)$
$f_1.Link(f_2, Rel("overlaps"))$		$(X^- < Y^-) \wedge (X^+ > Y^-) \wedge (X^+ < Y^+)$
$f_1.Link(f_2, Rel("overlappedby"))$		$(X^- > Y^-) \wedge (X^- < Y^+) \wedge (X^+ > Y^+)$
$f_1.Link(f_2, Rel("starts"))$		$(X^- = Y^-) \wedge (X^+ < Y^+)$
$f_1.Link(f_2, Rel("startedby"))$		$(X^- = Y^-) \wedge (X^+ > Y^+)$
$f_1.Link(f_2, Rel("during"))$		$(X^- > Y^-) \wedge (X^+ < Y^+)$
$f_1.Link(f_2, Rel("contains"))$		$(X^- < Y^-) \wedge (X^+ > Y^+)$
$f_1.Link(f_2, Rel("finishes"))$		$(X^- > Y^-) \wedge (X^+ = Y^+)$
$f_1.Link(f_2, Rel("finishedby"))$		$(X^- < Y^-) \wedge (X^+ = Y^+)$
$f_1.Link(f_2, Rel("equals"))$		$(X^- = Y^-) \wedge (X^+ = Y^+)$

Après avoir récupéré les tiers concernées par la recherche, il faut fixer une valeur de radius appropriée, qui dépend du type d'annotation. Cette étape favorisera les comparaisons, en s'appuyant sur les equations décrites en (1), dans les relations de la table 1. La valeur de temps s'indique en secondes :

```
gestures.SetRadius(0.02)
tokens.SetRadius(0.001)
```

Ensuite, une fonction booléenne est créée pour chacune des tiers. Dans cet exemple, il n'y a pas de critère de sélection sur les gestes. Par contre, il faut sélectionner les pauses dans la tier des tokens. Nous garderons les pauses silencieuses de plus de 200ms et les pauses pleines de plus de 100ms. Ces fonctions booléennes sont données en paramètre pour créer les filtres :

```
fgest = gestures()
ftokens = tokens((Bool(exact='euh') & Bool(duration_gt=0.1)) |\
          (Bool(exact='#') & Bool(duration_gt=0.2)))
```

Il faut alors déclarer une fonction de relations, qui dresse l'inventaire de chacune des relations de Allen qui pourra être vraie pour qu'une annotation soit acceptée :

```
relation = Rel('starts') | Rel('startedby') | Rel('finishes') | Rel('finishedby') |\
           Rel('during') | Rel('contains') | Rel('overlaps') | Rel('overlappedby')
```

La fonction de relation sert à créer un nouveau filtre pour obtenir une tier qui ne contient que les gestes recherchés :

```
f = fgest.Link(ftokens, relation)
new_gest_tier = fget.Filter()
```

6 Conclusion

Nous avons proposé un système d'extraction de données multi-annotées provenant, éventuellement, de différents logiciels d'annotation couramment utilisés par les linguistes. Ce système repose que l'implémentation de deux fonctions : une fonction booléenne pour filtrer sur le contenu des annotations et une fonction de relation basée sur l'algèbre de Allen. L'avantage principal de notre proposition réside dans le fait que *les requêtes peuvent être formulées par les annotateurs eux-mêmes, directement sur leurs annotations*, notamment via l'interface graphique. Ce système permet ainsi de répondre concrètement à un besoin exprimé par les annotateurs, qui sont dans l'impossibilité d'utiliser les systèmes de requêtes existant (SQL, Xquery, etc). Le système proposé fait parti de SPPAS, logiciel librement distribué. De plus, avec des compétences élémentaires en Python, les données peuvent également être interrogées à l'aide de l'API fournie. Dans un futur proche, il est notamment prévu d'ajouter l'importation d'annotations provenant d'autres logiciels.

Remerciements

Ce travail, mené dans le cadre de la mise en place de l'Equipex ORTOLANG a bénéficié d'une aide de l'État gérée par l'ANR au titre du programme Investissement d'Avenir portant la référence ANR-11-EQPX0032.

Références

BARRAS, C., GEOFFROIS, E., WU, Z. et LIBERMAN, M. (1998). Transcriber : a free tool for segmenting, labeling and transcribing speech. *In First International Conference on Language Resources and Evaluation*, pages 1373–1376, Granada (Spain).

BIGI, B. et HIRST, D. (2012). SPeech Phonetization Alignment and Syllabification (SPPAS) : a tool for the automatic analysis of speech prosody. *In* TONGJI UNIVERSITY PRESS, I., éditeur : *Speech Prosody*, Shanghai (China).

BIGI, B., WATANABE, T. et PRÉVOT, L. (2014). Representing multimodal linguistics annotated data. *In Language Resources and Evaluation Conference*, Reykjavik (Iceland).

BLACHE, P., BERTRAND, R., BIGI, B., BRUNO, E., CELA, E., ESPESSER, R., FERRÉ, G., GUARDIOLA, M., HIRST, D., MAGRO, E.-P., MARTIN, J.-C., MEUNIER, C., MOREL, M.-A., MURISASCO, E., NESTERENKO, I., NOCERA, P., PALLAUD, B., PRÉVOT, L., PRIEGO-VALVERDE, B., SEINTURIER, J., TAN, N., TELLIER, M. et RAUZY, S. (2010). Multimodal annotation of conversational data. *In The Fourth Linguistic Annotation Workshop*, pages 186–191, Uppsala (Sueden).

BOERSMA, P. (2001). Praat, a system for doing phonetics by computer. *Glot International*, 5 :9/10:341–345.

BRUGMAN, H., RUSSEL, A. et NIJMEGEN, X. (2004). Annotating multi-media / multimodal resources with ELAN. *In Fourth International Conference on Language Resources and Evaluation*, pages 2065–2068, Lisbon (Portugal).

KIPP, M. (2013). *ANVIL : A Universal Video Research Tool*. J. Durand, U. Gut, G. Kristofferson (Hrsg.) Handbook of Corpus Phonology, Oxford University Press.

LLISTERRI, J. (2011). Speech analysis and transcription software. http ://liceu.uab.es/˜joaquim/phonetics/fon_anal_acus/herram_anal_acus.html.

O'HALLORAN, K. L. (2011). Multimodal discourse analysis. *Companion to Discourse*, pages 120–137.

O'HALLORAN, K. L. et SMITH, B. A. (2012). Multimodal text analysis. *The Encyclopedia of Applied Linguistics.*

PUJARI, A. K., KUMARI, G. V. et SATTAR, A. (2000). INDU : An interval duration network. *In Sixteenth Australian joint conference on AI*, pages 291–303. SpringerVerlag.

RAUZY, S. et BLACHE, P. (2009). Un point sur les outils du LPL pour l'analyse syntaxique du français. *In Actes de la Journée ATALA "Quels analyseurs syntaxiques pour le français ?"*, Paris (France).

SJÖLANDER, K. et BESKOW, J. (2000). WAVESURFER - an open source speech tool. *In 6th International Conference of Spoken Language Processing*, Beijing (China).

Annotation de la temporalité en corpus : contribution à l'amélioration de la norme TimeML

Anaïs Lefeuvre[1], Jean-Yves Antoine[1], Agata Savary[1], Emmanuel Schang[2], Lotfi Abouda[2], Denis Maurel[1], Iris Eshkol[2]
(1) Université François Rabelais de Tours, laboratoire LI
(2) Université d'Orléans, laboratoire LLL, UMR 7270

Résumé. Cet article propose une analyse critique de la norme TimeML à la lumière de l'expérience d'annotation temporelle d'un corpus de français parlé. Il montre que certaines adaptations de la norme seraient conseillées pour répondre aux besoins du TAL et des sciences du langage. Sont étudiées ici les questions de séparation des niveaux d'annotation, de délimitation des éventualités dans le texte et de l'ajout d'une relation temporelle de type associative.

Abstract. This paper reports a critical analysis of the TimeML standard, in the light of a temporal annotation that was conducted on spoken French. It shows that the norm suffers from weaknesses that must be corrected to fit the needs of NLP and corpus linguistics. These limitations concern mainly 1) the separation of different levels of linguistic annotation, 2) the delimitation in the text of the events, and 3) the absence of a bridging temporal relation in the norm.

Mots-clés : annotation temporelle, TimeML, éventualités, relations temporelles, expressions temporelles
Keywords: temporal annotation, TimeML, eventualities, temporal relations, time expressions

1 Introduction : normalisation des ressources linguistique en TAL

La lecture des actes de la première édition de la conférence TALN, organisée il y a 20 ans à Marseille, nous renseigne sur l'évolution de la discipline, marquée par le développement des approches par apprentissage : seules 13% des communications de l'époque évoquaient des approches centrées sur les données. Cette évolution fut celle du passage de la linguistique computationnelle à l'ingénierie des langues. D'un point de vue méthodologique, elle s'est traduite par le recours de plus en plus systématique aux corpus. Avec l'émergence d'Internet et du *Big Data*, une masse sans cesse accrue d'information textuelle est désormais disponible. La question qui se pose n'est plus l'accès au matériau linguistique mais à son annotation, description enrichie essentielle à de nombreuses applications. L'annotation manuelle de corpus étant coûteuse, la création de telles ressources reste limitée. Pour palier cette difficulté, certains s'en remettent à une annotation automatique dont les erreurs sont compensées par la masse de données enrichies (Clark et Lappin 2010). La diffusion d'annotations manuelles d'envergure et de qualité reste néanmoins une question cruciale. Afin de favoriser la réutilisabilité des ressources, il est nécessaire de définir des normes d'annotation acceptées de tous. Cet effort de normalisation reste toutefois insuffisant. La reconnaissance des entités nommées est symptomatique de cette situation. La notion d'entité nommée n'a cessé d'évoluer au fil des campagnes d'évaluation sans arriver à un réel standard consensuel. D'où l'existence de corpus reposant sur des typologies peu compatibles.

L'annotation en temporalité semble avoir évité cet écueil. Elle a fait l'objet de la publication d'une norme, TimeML proposée par le comité TC37de l'ISO (2008). Elle a été adoptée par de multiples corpus, avec parfois des adaptations comme ce fut le cas pour le French TimeBank (Bittar and al., 2011). Cette norme présente toutefois à nos yeux comme à d'autres (Battistelli, 2009) des insuffisances qu'il convient de circonscrire. Après un état de l'art sur l'annotation en temporalité, cet article présente une analyse critique de certaines faiblesses de la norme ISO TimeML et tente d'y apporter des réponses. Cette réflexion est portée par un consortium dont l'ambition est de réaliser un corpus de grande envergure annoté en temporalité, comme il l'a déjà fait pour l'annotation en coréférence avec le corpus ANCOR_Centre (Muzerelle et al. 2013, Muzerelle et al. 2014). Ce texte ne constitue pas la proposition d'une nouvelle norme, mais le fruit de réflexions issues d'ateliers d'annotation que nous portons auprès de la communauté TALN pour alimenter le début d'une nécessaire évolution de la norme TimeML.

2 Annotation en temporalité

L'annotation de la temporalité linguistique demande dans un premier lieu de définir ce qu'est un évènement, élément constitutif de la temporalité du discours. On retiendra deux définitions pertinentes, la première de Mani et al. (2005) :

> *"We consider "events" a cover term for situations that happen or occur. Events can be punctual or last for a period of time. We also consider as events those predicates describing states or circumstances in which something obtains or holds true."* (Mani et al., 2005)

> "*Nous utiliserons (...) le terme d'éventualité (...) comme terme générique pour désigner quelque chose qui se produit, qui a lieu ou qui est vrai sur une période donnée de temps*." (De Saussure, 1997)

Ces deux définitions rendent compte de la diversité des évènements évoqués dans le discours. Diversité que nous chercherons à représenter par le terme générique d'éventualité (Bach, 1981), et non d'évènement. Pour analyser temporellement un énoncé, il est également nécessaire de détecter les relations entre ces entités du discours. (Allen, 1983) en propose un modèle standard reposant sur 13 relations minimales entre intervalles.

(1) *Tu restes la semaine à Paris, j'y serai à partir de mercredi*.
(2) *Je reste la semaine à Paris d'où je pars pour Londres*.

L'exemple (1) illustre la relation d'*overlap*, qui suppose l'existence d'une intersection entre les intervalles mobilisés par "*tu restes la semaine à Paris*" et "*j'y serai à partir de mercredi*". L'exemple (2) illustre la relation *meet*. Celle-ci décrit le fait qu'une éventualité, ("*je pars pour Londres*") succède immédiatement à une autre ("*je reste la semaine à Paris*").

L'annotation de la temporalité a été initialement motivée par des visées applicatives, notamment en extraction d'information. Des campagnes d'évaluation ont défini des tâches bien délimitées, de complexité croissante, chacune étant associée à un schéma d'annotation spécifique. Ainsi, MUC-7 et CoNLL 2002-2003 ont proposé les tâches de repérage d'entités nommées incluant des expressions temporelles. ACE 2005-2007 a complété cette tâche par une phase de normalisation d'expressions temporelles et a introduit la tâche de repérage et de caractérisation d'évènements. TempEval à son tour a proposé une tâche d'identification automatique des relations temporelles à partir des expressions temporelles et événements déjà identifiés dans le texte. Un des résultats de ces initiatives est la construction progressive d'un standard d'annotation qui a donné lieu à la norme TimeML, dont nous donnons un aperçu à la section suivante.

Un des challenges récents dans la reconnaissance d'entités nommées, y compris temporelles, est celui du repérage des entités imbriquées (Savary et al. 2010, Gravier et al., 2012), comme par exemple dans l'entité nommée [*conseil général* [*du Morbihan*]]. L'imbrication permet de mettre directement en évidence certaines relations inter-entités, mais complexifie la tâche des systèmes de reconnaissance automatique. Certains travaux tentent de contourner la difficulté du repérage des frontières de ces entités/mentions en identifiant seulement leurs têtes lexicales. Comme nous l'expliquerons dans la section 3.2, nous nous plaçons du côté des approches qui délimitent la portée textuelle complète des entités/mentions (Muzerelle et al. 2013, Ogrodniczuk et al. 2013).

Notons enfin, que l'intérêt de la prise en compte de l'information temporelle est partagée par d'autres domaines. Ainsi, les spécialistes de représentation de connaissances, et notamment des Linked Open Data, ont identifié le besoin de raisonnement automatique de nature spatio-temporelle, et ont pour cela proposé des extensions aux ontologies existantes, telles que YAGO2 (Hoffart et al., 2011), et aux formalismes associés (RDF, SPARQL, etc.), en y incluant notamment des dimensions temporelles (par addition d'intervalles temporels pour caractériser l'existence des entités).

3 TimeML

Dans le domaine du TALN, la norme ISO TimeML s'est imposée comme un stantard de facto pour l'annotation de la temporalité. Des corpus annotés suivant le format ISO TimeML existent ainsi pour l'anglais, le français, l'italien, le portugais, le coréen, le roumain et le chinois. TimeML distingue tout d'abord trois types d'unités:

- EVENT, reprenant globalement la notion d'éventualité typée (*state*, *occurrence*, etc.) et attachée à la tête du syntagme (verbal, nominal, adjectival ou prépositionnel) correspondant. Un EVENT est assorti d'un ou plusieurs MAKEINSTANCE qui permettent d'annoter les différentes instances du même EVENT en donnant plusieurs traits à ceux-ci (part-of-speech, tense, aspect, etc.). Par exemple, l'énoncé suivant sera annoté avec un EVENT attaché à "*enseigné*" et deux objets MAKEINSTANCE, un pour chacune des instances de cette éventualité.

  (3) *Jean a enseigné deux fois lundi*

- TIMEX : les expressions temporelles expriment une date calendaire, un horaire, une durée, un ensemble d'horaires (*date*, *time*, *duration*, *set*). Pour l'exemple (3), "*lundi"* sera annoté TIMEX.
- SIGNAL : termes linguistiques qui marquent l'existence d'une relation temporelle entre deux éventualités.

Enfin, 3 relations sont proposées :

- TLINK : relation entre deux EVENTs, un EVENT et un TIMEX, ou encore entre deux TIMEXs, suivant la typologie de Allen. TimeML ajoute une relation *identity* pour la coréférence temporelle.
- ALINK : relation aspectuelle entre EVENTs, permet d'annoter l'opération d'un EVENT sur un second telle que la sélection de sa phase initiale, finale, son apogée etc. Dans l'exemple suivant, "*démarré"* et "*cours"* sont deux EVENTs reliés par une relation ALINK dont le type est *initiates* :

  (4) *Jean a démarré son cours à 14h.*

- SLINK : relation de subordination entre deux EVENTs, marquant l'influence du premier sur la factivité du second. Dans l'exemple suivant "*oublié"* et "*était à Paris"* sont liés par un SLINK de type *factive*.

  (5) *Jean a oublié qu'il était à Paris ce jour là.*

4 Limites de TimeML

Dans le cadre du projet TEMPORAL, financé par la MSH Val de Loire, nous comptons ajouter une couche d'annotation temporelle au corpus ANCOR_Centre (Muzerelle et al. 2013, 2014). Couvrant un large éventail de relations anaphoriques ou de coréférence, ce grand corpus oral ne recense toutefois pas les anaphores abstraites (Dipper & Zinsmeister, 2010) telles que :

(6) L1 : *Pierre a encore cassé sa voiture.*
L2 : *Venant de lui, **ça** ne m'étonne pas.*

Ici, le pronom anaphorique "*ça"* a pour antécédent l'éventualité décrite par l'ensemble de l'énoncé de L1. TEMPORAL avait pour objectif initial de rendre compte de ce genre de situations avec la norme TimeML. Une démarche méthodologique consistant à réaliser à intervalles réguliers des ateliers d'annotation entre experts nous a toutefois révélé certaines limitations pénalisantes de la norme, qui sont l'objet du présent article.

4.1 Confusion entre les niveaux d'annotation

ISO TimeML cherche visiblement à intégrer dans une seule couche d'annotation tous les éléments nécessaires à la résolution des références temporelles. Ce choix conduit à des corpora auto-suffisants mais induit des incohérences du fait de la juxtaposition d'annotations de niveaux linguistiques différents. C'est ainsi le cas de la distinction entre évènement et instance (EVENT/MAKEINSTANCE). EVENT est en effet directement lié à un observable linguistique, marqueur de l'éventualité dans l'énoncé, tandis que l'instance réfère à la réalisation effective de cette éventualité. Paradoxalement, TimeML confère à l'instance l'ensemble des traits qui décrivent l'éventualité, alors que ceux-ci sont clairement liés à sa réalisation linguistique en contexte (*tense*, *aspect*, *etc*.).

(7) *Jean a enseigné deux fois lundi.*

Ainsi, l'exemple (7), transposé du guide d'annotation TimeML, montre une répétition d'éventualités. La norme propose de définir un EVENT pour décrire une éventualité traduite par le verbe "*a enseigné"*, puis deux instances correspondant aux deux réalisations successives de cette éventualité. Il serait préférable de définir des niveaux d'annotation séparés, ce que permet le format déporté d'ISO TimeML. Nous conseillons même de ne conserver que les objets EVENT en leur intégrant les attributs de MAKEINSTANCE comme dans le French TimeBank (Bittar et al. 2011), nous limitant à une annotation objective purement linguistique. L'annotation combinée des deux niveaux pose en effet des questions en termes de fiabilité des données. Considérons maintenant l'énoncé (8) :

(8) *Jean a enseigné aux L1 le lundi et Marc aux L2 le mardi.*

Doit-on encore considérer comme TimeML qu'il n'y a qu'une seule éventualité et plusieurs instances, alors que les actions diffèrent, puisque les arguments du verbe varient ? Dès lors, où s'arrête la distinction entre éventualités et instances ? Dans le cadre du projet TEMPORAL qui réunit nos deux laboratoires, nous proposons de nous limiter à une annotation purement linguistique qui limite ce genre de questionnement. L'annotation pourrait ensuite être complétée par d'autres couches pour les communautés travaillant sur la recherche d'information ou les systèmes question/réponse.

4.2 Délimitation des éventualités

La remarque précédente pose la question de la délimitation des éventualités. ISO TimeML a choisi de caractériser les éventualités uniquement par la tête lexicale de leur observable linguistique, en ignorant leurs arguments. Ce choix est motivé par le souci de simplifier la tâche des annotateurs (Bittar, 2008). Cette question a préocuppé Pustejovsky et ses collègues, qui ont proposé d'annoter à part les arguments des têtes lexicales portant une éventualité (Pustejovsky et al. 2006). Cette proposition n'a toutefois pas été retenue dans la norme TimeML.

Nos expériences d'annotation n'ont pas démontré de difficulté à délimiter des éventualités à large empan. Il nous semble donc préférable d'annoter l'ensemble de l'unité linguistique décrivant l'éventualité. En effet, le calcul de la temporalité d'un énoncé se fait à partir de tous ses éléments pertinents, ce dont doit rendre compte le balisage dû à l'annotation. Cette approche permet de se dispenser des relations artefactuelles auxquelles TimeML est condamné dans le cas d'encapsulation d'éventualités comme dans l'exemple (9) :

(9) [*vous voulez* [*venir avec votre chien*]]

Ce choix est cohérent avec les pratiques observées pour l'annotation des entités nommées. Il permet d'établir une procédure homogène et syntaxiquement cohérente avec le traitement des éventualités nominales. L'exemple ci-dessous fait ainsi un parallèle entre les délimitations d'entités imbriquées proposées pour notre annotation temporelle (10a) et pour une annotation en entités nommées (10b) telle qu'envisagée par exemple dans la campagne ETAPE.

(10a) [*le début de* [*la seconde phase de* [*colonisation de l'archipel*]]]
(10b) [*le président de* [*la première réunion de* [*la nouvelle assemblée constituante*]]]

L'intérêt de ce choix est manifeste dans le cas des anaphores abstraites ou de l'encapsulation d'éventualités. Dans l'exemple (11), l'éventualité ancrée par *vouloir* a une portée sur les deux autres éventualités, ce qui est aisément capturé par l'encapsulation. TimeML peut représenter ce type d'énoncé, mais d'une manière moins élégante avec des relations.

(11) [*vous voulez* [*venir avec votre chien*] *et* [*faire l'intégralité de la promenade*]]

Il n'en reste pas moins qu'une annotation à large empan peut conduire à de nouvelles difficultés que nous avons-nous-même expérimentées. Tout d'abord, les relations entre les têtes lexicales et leurs arguments peuvent s'étendre sur de grandes longueurs. Nous proposons de nous limiter à l'inclusion des éléments sous-catégorisés (arguments de valence pour un verbe, par exemple). D'autre part, les éventualités ainsi délimitées peuvent être discontinues. C'est le cas à l'oral où des disfluences telles que les incises peuvent découper l'éventualité. On peut formellement décrire ces discontinuités (notion de schéma sous la plateforme Glozz, par exemple (Mathet & Widlöcher, 2009)) mais cela complique la tâche d'annotation. C'est pourquoi notre réflexion s'oriente à l'heure actuelle vers une annotation sur treebank, inspirée de l'annotation des expressions polylexicales dans le Prague Dependency Treebank (Bejček & Straňák, 2010): puisqu'une éventualité est représentée dans un treebank par le nœud de l'arbre syntaxique qui correspond à la tête lexicale tout en couvrant la partie de l'énoncé concerné, identifier cette éventualité se réduirait à pointer ce noeud dans l'arbre tout en lui assignant des attributs spécifiques à la temporalité.

Des incohérences plus anecdotiques de délimitation peuvent également se rencontrer. TimeML distingue ainsi deux types d'entités qui modifient ou précisent la portée temporelle des éventualités ou des expressions temporelles :

- les signaux, qui indiquent la présence d'une relation temporelle (TLINK) entre deux éventualités ou expressions temporelles. Ainsi, la préposition *avant* sera annotée comme un signal dans l'énoncé *je suis parti* ***avant*** *midi*.
- les modifieurs (MOD), qui altèrent ou précisent une portée temporelle. Par exemple, la locution *en fin de* sera annoté comme un modifieur dans l'énoncé *je suis parti* ***en fin de*** *matinée*.

Ces entités jouent des rôles relativement proches mais donnent lieu à des conventions d'annotation radicalement différentes. Les signaux sont des objets d'annotation propres, délimités par un balisage spécifique (`<SIGNAL> avant </SIGNAL>`) alors que les modifieurs conduisent simplement à l'ajout d'un attribut MOD dans l'élément qu'ils caractérisent. ISO TimeML donnera ainsi sur notre exemple l'annotation (simplifiée) suivante : `en fin de <EVENT … MOD='END'> matinée </EVENT>`. Une description homogène rendrait plus fiable le processus d'annotation.

4.3 Expressions temporelles

L'annotation des expressions temporelles dans TimeML dérive du tag TIMEX2 utilisé dans TIDES (Ferro 2005). Cette norme répond plus aux besoins de la RI que du TAL. Les expressions temporelles y sont en effet intégralement évaluées temporellement en contexte. Ainsi, "*le mois dernier*" sera par exemple annotée `2014-05`. Cette convention pose des

problèmes de cohérence : le contexte d'évaluation ne sera en effet pas toujours connu pour permettre cette résolution temporelle. Cela renforce par ailleurs la part de subjectivité laissée à l'annotateur, qui ne cherchera pas toujours l'information nécessaire à cette évaluation. Ici encore, TimeML mélange deux niveaux d'annotation : celui du signifiant linguistique et celui de son interprétation qui relève d'un raisonnement purement temporel. Nous proposons donc d'annoter "*le mois dernier"* sous la forme -1M (mois antérieur) et de laisser à une autre couche d'annotation la question de la résolution complète de l'expression temporelle.

4.4 Attributs liés aux éventualités

Enfin, certaines limitations concernent les attributs liés aux éventualités. L'une d'entre elles est spécifique l'adaptation au français de TimeML (Bittar et al. 2011). Le French TimeBank ajoute, pour le français, des temps dans le système de valeurs de l'attribut TimeML `tense`. On retrouve un temps `past` mais également un temps `imperfect`. Ce choix manque de pertinence linguistique : le terme *imperfect* étant traditionnellement attaché à une notion d'aspect, les valeurs du trait ici croisent d'un côté une notion de temps linguistique (*tense* en anglais et qui pourrait justifier `imperfect`) et de l'autre une représentation du temps extralinguistique en 3 parties `past, present, future`.

D'autres limitations correspondent à des manques de la norme. Par exemple ISO TimeML prévoit un attribut `neg` qui permet de représenter la négation d'une éventualité (exemple : *il ne pleut pas*). Dans cet esprit, il nous semble manquer un attribut pour marquer l'occurrence de l'éventualité dans une question. Cette information peut par exemple avoir une utilité pour l'identification des actes de langage liés à l'éventualité. Trois valeurs peuvent être portées par cet attribut :

- `YEST` si la question porte sur la temporalité de l'évènement. Par exemple : *quand viendras-tu ?*
- `YES` si la question porte sur l'occurrence ou non de l'éventualité. Par exemple : *viendras-tu ?*
- `NO` lorsqu'il n'y a pas d'une question ou qu'elle porte sur une autre dimension. Par exemple : *Où vas-tu ?*

4.5 Relations temporelles

Le dernier élément important de la norme concerne les relations temporelles. ISO TimeML intègre un type de relation (TLINK) qui peut concerner de manière indifférente les éventualités ou les expressions temporelles, alors que les types ALINK et SLINK ne concernent que les éventualités. Nous regrettons là encore le groupement sous la même annotation d'objets décrivent des réalités différentes (cf § 3.1.). Il serait plus cohérent de réserver le type TLINK aux seules relations entre éventualités, et d'ajouter un type spécifique pour les relations d'ancrage liant une expression temporelle à une éventualité. Pustejovsky suit d'ailleurs cette logique lorsqu'il propose (séminaire invité, Paris 2014) de créer une relation MLINK pour représenter une relation entre un TIMEX et un EVENT dont le premier donne la durée du second.

On relève en outre que le type TLINK se limite aux relations d'ordonnancement (égalité, précédence, post-occurence...) de Allen (1983). Cette convention ignore l'existence de correspondances plus complexes, comme par exemple des relations d'inclusion temporelle ontologiquement décrites. Considérons l'exemple suivant :

(12) *Je suis né à Noël, précisément, le 25 décembre 1966. Cette année-là il a fait très froid durant tout l'hiver*

TimeML représente *25 décembre 1966* comme une date mais ne peut exprimer la relation d'inclusion qu'elle partage ontologiquement avec l'expression englobante *cette année-là* ? On se retrouve ici avec une relation de type meronymique (`partie_de`) au niveau temporel. Celle-ci s'apparente par analogie avec les anaphores associatives que l'on retrouve lorsque l'on s'interesse à la référence (Webber 1988). L'attribut `reltype` de TLINK doit décrire ce type de relation. Par analogie, on propose de rajouter la valeur `bridging` au système de valeur associé à cet attribut.

5 Conclusion

Cet article a cherché à caractériser certaines limitations de la norme ISO TimeML qui demandent à être corrigées pour atteindre une annotation qui réponde réellement aux besoins du TALN, des sciences du langage mais également de l'ingénierie des connaissances. Certaines réponses à ces observations ne requièrent qu'une modification à la marge de la norme. La question de la délimitation des éventualités étant plus problématique de notre point de vue. Nous proposons de ne pas se limiter à la tête lexicale mais bien de l'étendre à tout le signifiant linguistique correspondant. La taille des ressources annotées avec la norme TimeML reste encore limitée. Il nous semble donc plus qu'urgent de la remettre partiellement en question, sur des points qui, comme nous l'avons rappelé, ont souvent été déjà questionnés.

Références

ALLEN J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11), 832–843.

BACH E. W. (1981). Time, Tense, and Aspect : An Essay in English Metaphysics . In Radical Pragmatics, 63–81.

BATTISTELLI D. (2009). *La temporalité linguistique: circonscrire un objet d'analyse ainsi que des finalités à cette analyse* Habilitation à diriger des Recherches, Université de Nanterre-Paris X.

BEJČEK E. & STRAŇÁK P. (2010). Annotation of multiword expressions in the Prague Dependency Treebank. *Language Resources and Evaluation*, 44(1–2). 7–21.

BITTAR A. (2008) Annotation des informations temporelles dans des textes en français. Actes *RECITAL'2008*.

BITTAR A., AMSILI P., DENIS P., DANLOS L. (2011). French TimeBank: An ISO-TimeML Annotated Reference Corpus. Proc. *ACL'2011,* Portland, Oregon : États-Unis.

CLARK A. & LAPPIN S. (2010). Unsupervised learning and grammar induction. *The Handbook of Computational Linguistics and Natural Language Processing*, 57.

DE SAUSSURE L. (1997). Le temps chez Beauzée : algorithmes de repérage et comparaison avec Reichenbach. *Cahiers Ferdinand de Saussure*, (49):171–195.

DIPPER S., ZINSMEISTER H. (2010). Towards a standard for annotating abstract anaphora. In *LREC 2010 Workshop on Language Resources and Language Technology Standards*, 54-59.

FERRO L. (2005). TIDES 2005 standard for the annotation of temporal expressions. *MITRE Corp. Tech. Report.*

GRAVIER G., ADDA G., PAULSON N., CARRÉ M., GIRAUDEL A. & GALIBERT O.(2012). The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. In International Conference on Language Resources, Evaluation and Corpora, na. Turquie.

ISO (2008). ISO DIS 24617-1: 2008 Language resource management - Semantic annotation framework - Part 1: Time and Events. International Organization for Standardization, ISO Central Secretariat, Genève, Suisse.

HOFFART J., SUCHANEK F. M., BERBERICH K., LEWIS-KELHAM E., DE MELO G. & WEIKUM G. (2011). YAGO2: exploring and querying world knowledge in time, space, context, and many languages, in Proc. *20th International Conference on World Wide Web, WWW 2011*, Hyderabad, India (Companion Volume), 229–232.

MANI I., PUSTEJOVSKY J., GAIZAUSKAS R., SAURI R., CASTANO J., LITTMAN J., SETZER A., & KATZ G. (2005). The specification language TimeML. In Mani I., Pustejovsky J., Gaizauskas R. (Eds.), *The language of time : a reader*, chapter 27. Oxford.

MATHET, Y., WIDLÖCHER, A. (2009). La plate-forme GLOZZ : environnement d'annotation et d'exploration de corpus. *Actes de TALN-2009*, pages 1–10

MUZERELLE J., LEFEUVRE A., ANTOINE J.-Y, SCHANG E, MAUREL D., VILLANEAU J., ESHKOL I. (2013). ANCOR, premier corpus de français parlé d'envergure annoté en coréférence et distribué librement. Actes *TALN'2013*, Les Sables d'Olonnes.

MUZERELLE J., LEFEUVRE A., SCHANG E., ANTOINE J.-Y., PELLETIER A., MAUREL D., ESHKOL I., VILLANEAU J. (2014). ANCOR_Centre, a large free spoken french coreference corpus. Proc. *LREC'2014*, Reykjavik.

OGRODNICZUK M., ZAWISŁAWSKA M., SAVARY A., GŁOWŃSKA K. (2013). Coreference_Annotation Schema for an Inflectional Language, Proc. *CICLING'2013*, Samos, Greece, In. *LNCS 7816*, Springer Verlag, 394-407.

PUSTEJOVSKY J., LITTMAN J., SAURI R. (2006). Argument Structure in TimeML. In Katz G., Pustejovsky J., Schilder F. (Eds.) *Annotating, Extracting and Reasoning about Time and Events*, Dagstuhl Seminar Proc., Schloss Dagstuhl, Allemagne.

SAVARY A., WASZCZUK J., PRZEPIÓRKOWSKI A. (2010). Towards the Annotation of Named Entities in the National Corpus of Polish. in Proc. *LREC'10*, Valetta, Malta.

WEBBER B. L. (1988). Tense as discourse anaphor. *Computational Linguistics*, *14*(2), 61-73.

Identification automatique de zones dans des documents pour la constitution d'un corpus médical en français

Louise Deléger Aurélie Névéol
LIMSI – CNRS UPR 3251, Orsay, France
louise.deleger@limsi.fr, aurelie.neveol@limsi.fr

Résumé. De nombreuses informations cliniques sont contenues dans le texte des dossiers électroniques de patients et ne sont pas directement accessibles à des fins de traitement automatique. Pour pallier cela, nous préparons un large corpus annoté de documents cliniques. Une première étape de ce travail consiste à séparer le contenu médical des documents et les informations administratives contenues dans les en-têtes et pieds de page. Nous présentons un système d'identification automatique de zones dans les documents cliniques qui offre une F-mesure de 0,97, équivalente à l'accord inter-annoteur de 0,98. Notre étude montre que le contenu médical ne représente que 60% du contenu total de notre corpus, ce qui justifie la nécessité d'une segmentation en zones. Le travail d'annotation en cours porte sur les sections médicales identifiées.

Abstract. Much clinical information is contained in the free text of Electronic Health Records (EHRs) and is not available for automatic processing. To advance Natural Language Processing of the French clinical narrative, we are building a richly annotated large-scale corpus of French clinical documents. To access the most medically relevant content of EHRs we develop an automatic system to separate the core medical content from other document sections, such as headers and footers. The performance of automatic content extraction achieves 96.6% F-measure, on par with human inter-annotator agreement of 98%. We find that medically relevant content covers only 60% of clinical documents in our corpus. Future annotation work will focus on these sections.

Mots-clés : Traitement Automatique de la Langue Biomédicale, segmentation de documents, identification de zones.

Keywords: BioNLP, Automatic document segmentation, section identification.

1 Introduction

De nombreuses informations cliniques sont contenues dans le texte des dossiers électroniques de patients et ne sont pas directement accessibles à des fins de traitement automatique. Afin de faciliter le développement d'outils et de méthodes de traitement automatique de la langue clinique, nous préparons un large corpus de documents cliniques annotés en entités biomédicales et en relations. Une première étape de ce travail consiste à séparer le contenu médical des documents et les informations administratives contenues par exemple dans les en-têtes et pieds de page. En effet, de nombreux documents de notre corpus contiennent des passages indiquant les coordonnées du service hospitalier ayant produit le document, ainsi qu'une liste des médecins susceptibles d'intervenir dans le service. Ces informations ne sont pas spécifiques au patient ni aux éléments de la prise en charge décrits dans le document. Dans cet article, nous présentons la typologie en zones de documents que nous avons établie pour notre corpus clinique ainsi qu'une méthode pour automatiquement identifier ces zones et isoler le contenu médical des documents contenus des dossiers électroniques de patients.

La structure discursive d'un document peut être très utile pour améliorer des outils de recherche d'information (Ruch *et al.*, 2007). Ainsi, de nombreux travaux ont cherché à détecter automatiquement la structure d'articles scientifiques. Certaines approches ont catégorisé les phrases de résumés scientifiques en sections telles que "Introduction", "Method", "Result", "Conclusion" (McKnight & Srinivasan, 2003; Yamamoto & Takagi, 2005; Lin *et al.*, 2006; Ruch *et al.*, 2007; Hirohata *et al.*, 2008). D'autres ont travaillé sur l'intégralité des articles en identifiant des zones argumentatives (définies par Teufel (2000)) du type "Background" (contexte), "Own" (description des travaux des auteurs), "Other" (description d'autres travaux), etc. (Teufel & Moens, 2002; Merity *et al.*, 2009). Différents types de classifieurs ont été développés pour ces tâches : Naive Bayes (Teufel & Moens, 2002; Ruch *et al.*, 2007), MaxEnt (Maximum Entropy) (Merity *et al.*, 2009), SVM (Support Vector Machines) (McKnight & Srinivasan, 2003; Yamamoto & Takagi, 2005), HMM (modèles

de Markov cachés) (Lin *et al.*, 2006), CRF (champs conditionnels aléatoires) (Hirohata *et al.*, 2008). Les performances les plus hautes atteignent des valeurs supérieures à 0.90 d'exactitude (accuracy) pour la classification des phrases, en particulier le modèle CRF de Hirohata *et al.* (2008) avec 0.943 d'exactitude.

Pour traiter les dossiers électroniques de patients, des approches ont été développées pour identifier les différentes sections qui structurent les documents cliniques, comme par exemple "Indication", "Antécédents médicaux", "Evolution dans le service", "Traitement de sortie", etc. (Denny *et al.*, 2009; Li *et al.*, 2010; Tepper *et al.*, 2012). Certaines études se sont limitées à la classification des sections, en présupposant la connaissance des frontières de sections. Par exemple, Denny *et al.* (2009); Li *et al.* (2010) utilisent des frontières de section obtenues grâce à des heuristiques établies manuellement, ce qui est peu généralisable à d'autres corpus. Dans un travail plus récent, Tepper *et al.* (2012) effectuent à la fois la détection des frontières de section et la classification du type de section, à l'aide de modèles HMM. Ils obtiennent des F-mesures proches de 0.92 sur leurs différents corpus de tests pour la détection et la classification des débuts de sections.

Enfin, les approches de segmentation en zones ont également été utilisées sur d'autres types de documents, comme par exemple le contenu textuel des e-mails (Lampert *et al.*, 2009).

Dans cette étude, nous proposons une méthode d'identification des zones de haut niveau (de type en-tête, pied de page, contenu principal) des textes cliniques, qui s'appuie sur les travaux de la littérature en détection de sections.

2 Matériel et méthodes

2.1 Documents cliniques

Pour cette étude, nous avons utilisé des documents d'un corpus de 138 000 textes cliniques issus d'un groupe d'institutions hospitalières françaises. Ce corpus contient environ 2 000 dossiers électroniques de patients. Il couvre de nombreuses spécialités médicales et plusieurs types de documents (principalement des courriers, des comptes rendus de séjour, des comptes rendus d'acte et des ordonnances). Nous avons constitué deux corpus de travail, comprenant des documents aléatoirement sélectionnés dans le corpus entier : un corpus d'entraînement et un corpus de test de 100 documents chacun. Par la suite, nous avons constitué un corpus destiné à l'annotation en entités et en relations (500 documents) à partir des documents du service "Hépato-gastro-nutrition", en nous appuyant sur le système développé. Comme nous avons pour objectif d'annoter ce corpus (ultérieurement), nous avons corrigé la segmentation effectué par notre système afin de ne manquer aucune ligne de contenu. Ceci nous a ainsi donné l'occasion d'évaluer notre système sur un corpus supplémentaire.

2.2 Accès au contenu discursif : identification de zones dans les documents

Nous avons établi un découpage en zones des documents hospitaliers et distinguons ainsi quatre types de zone :
– l'*en-tête générique* : de type « papier à lettre », avec les coordonnées du service dont provient le document. On observe le même en-tête pour tous les documents de même provenance.
– l'*en-tête spécifique* : contenant des informations utiles comme la date du jour, le lieu, le nom et la date de naissance du patient, etc. Ce type d'en-tête est spécifique à un document.
– le *contenu* principal du document
– le *pied de page* : signature du médecin, éventuellement accompagnée de formules de politesse s'il s'agit d'un courrier.

Le tableau 1 montre un exemple type de document découpé en zones. Il faut cependant remarquer que toutes les zones ne sont pas obligatoirement présentes dans chaque document ; il existe par exemple des documents sans en-tête générique ou sans pied de page. De même, certaines zones peuvent se retrouver à plusieurs endroits d'un même document. Par exemple, les « PS » apparaissant dans un courrier après la signature du médecin sont considérés comme des zones de contenu.

Afin de développer et d'évaluer notre méthode de découpage automatique en zones, le corpus d'entrainement et le corpus de test a été manuellement annoté par les deux auteurs (LD, AN). L'annotation a été effectuée avec le logiciel Brat Rapid Anotation Tool (BRAT) (Stenetorp *et al.*, 2012), en marquant le début de chacune des différentes zones selon notre typologie. Nous avons choisi ce mode d'annotation car il permet de représenter les zones simplement : une zone s'achève au début de la zone suivante ou à la fin du document. Les accords inter-annotateurs (F-mesure) pour chacun des corpus sont présentés dans le tableau 2. On observe que les accords sont hauts, avec une micro-moyenne de 0,8711 et 0,9625 respectivement pour le corpus d'entrainement et de test.

En-tête générique	Hôpital Deschamps - 15, avenue du général Leclerc 77000 Lyon |Service de Chirurgie Générale |Secrétariat (01.41.54.92.89 || |Pr Pierre LEBLOND |E-mail : Chirurgie.Digestive@deschamps.fr |Dr Marie MICHEL |
En-tête spécifique	COMPTE-RENDU : CONSULTATION du 21/07/2005 De : Marc DURAND Né(e) le : 14/04/1958 Paris le 22/07/2005 Mon cher confrère,
Contenu	Je vois ce jour en consultation, Monsieur Marc DURAND, opéré de condylômes au niveau de la marge anale et dans le canal anal le 24 novembre 2004. Les suites opératoires ont été simples. Actuellement, l'examen anal et péri-anal est normal, il n'y a plus de condylômes, les cicatrices sont propres. Dans ces conditions, il n'y a pas lieu de revoir Monsieur DURAND en consultation.
Pied de page	Bien confraternellement. J. DUPONT – Interne.

TABLE 1 – Exemple de document découpé en zones ; toutes les informations identifiantes ainsi que les dates ont été remplacées par des substituts plausibles

	Entrainement (N=100)	**Test (N=100)**
En-tête générique	0,9036	1,0000
En-tête spécifique	0,7411	0,9565
Contenu	0,9483	0,9384
Pied de page	0,9000	0,9622
Micro-moyenne	**0,8711**	**0,9625**

TABLE 2 – Accords inter-annotateurs pour le découpage en zones

Premier token de la ligne
Premier token de la ligne précédente
Premier token de la ligne suivante
Deuxième token de la ligne
Bigramme du 1er et 2è tokens de la ligne
Le premier token de la ligne est-il tout en majuscules ?
Position relative de la ligne dans le document (en cinquièmes)
Longueur de la ligne en nombre de tokens
Présence de lignes blanches avant la ligne
Présence de chiffres dans la ligne
Présence d'adresses emails dans la ligne

TABLE 3 – Ensemble de traits utilisés dans le modèle statistique. 'token' = unité de segmentation minimale

Une fois les désaccords résolus, le corpus d'entrainement a été utilisé pour mettre en place et améliorer notre système de découpage en zones. Le corpus de test a été utilisé comme jeu de test pour évaluer notre système final. Pendant la phase de développement, nous avons effectué des validations croisées (avec une partition en 10 ensembles) sur le corpus

d'entrainement afin d'optimiser l'ensemble de traits utilisés dans le modèle statistique.

Pour identifier automatiquement les zones des documents, nous avons entraîné un modèle statistique à champs conditionnels aléatoires (CRF (Lafferty *et al.*, 2001)) en utilisant l'outil Wapiti (Lavergne *et al.*, 2010). Nous classifions chaque ligne d'un document comme appartenant à l'un des quatres types de zones, en nous appuyant sur le format BIO (Beginning/Inside/Outside) pour distinguer les débuts de zones. Cette approche est similaire à celles proposées par Tepper *et al.* (2012) pour segmenter des textes cliniques en sections et Hirohata *et al.* (2008) pour identifier les sections d'articles scientifiques. L'ensemble des traits utilisés dans notre modèle CRF est détaillé dans le tableau 3.

Une fois le système développé, nous l'avons ré-entraîné sur l'ensemble des documents annotés à disposition (soit 200 documents). Nous l'avons appliqué sur le corpus entier (138 000 documents) pour sélectionner sur la base du contenu détecté automatiquement le corpus de 500 documents que nous planifions d'annoter en entités et relations. Nous avons corrigé le découpage en zones effectué par notre système, manuellement à l'aide de l'outil BRAT.

2.3 Evaluation

Nous avons évalué les performances de la méthode d'identification des zones en examinant d'abord les performances de la détection des débuts de zone, mesurées en terme de précision, rappel et F-mesure, pour chaque type de zone ainsi que la micro-moyenne. Comme nous cherchons à déterminer le contenu principal pour sélectionner des documents, il est important que les lignes de contenu soient identifiées avec un très bon rappel, et également une bonne précision. Nous avons donc également évalué les résultats pour l'ensemble des lignes, c'est-à-dire l'attribution d'une zone à chaque ligne des documents. L'évaluation de la classification des lignes a été mesurée en terme d'exactitude (accuracy) pour l'ensemble des lignes et en terme de précision, rappel et F-mesure pour chaque type de zone. L'évaluation de notre méthode a été effectuée sur deux corpus : (1) sur le corpus de test (100 documents) avec le système construit sur le corpus d'entraînement (100 documents) et (2) sur le corpus destiné à l'annotation (500 documents) avec le système construit sur la combinaison des corpus d'entraînement et de test (2x100 documents).

3 Résultats

3.1 Performances de l'identification des zones sur le corpus de test

Les performances de la détection des débuts de zone sont présentées dans le tableau 4, individuellement pour chaque type de zone ainsi que leur micro-moyenne. Les résultats sont très bons pour l'identification des débuts d'en-têtes (F-mesures supérieures à 0,94), et un peu plus bas pour les débuts de contenu et de pieds de page (F-mesures proches de 0,81). La micro-moyenne est de 0,88 de F-mesure, ce qui est légèrement supérieur à l'accord inter-annotateur sur l'échantillon 1 (0,87), mais inférieur à celui sur l'échantillon 2 (0,96). Les résultats de l'évaluation de la classification des lignes en zone sont données dans le tableau 5). L'exactitude (accuracy) est de 0,9678, et les performances sont particulièrement hautes pour les lignes de contenu et d'en-têtes.

	Précision	**Rappel**	**F-mesure**
Début d'en-tête générique	1,0000	0,9302	0,9639
Début d'en-tête spécifique	0,9697	0,9231	0,9458
Début de contenu	0,8384	0,7830	0,8098
Début de pied de page	0,8471	0,7742	0,8090
Micro-moyenne	**0,9118**	**0,8509**	**0,8803**

TABLE 4 – Performances de l'identification des débuts de zones (corpus de test)

Une analyse des erreurs montre que les frontières de section les plus difficiles à placer sont celles concernant le début du contenu et le début du pied de page. Dans les courriers (cf. table 1), la formule de politesse qui marque le début de la zone de pied de page apparait parfois directement à la suite du dernier élément de contenu (par exemple, "Je reste à votre disposition pour revoir Mr DURAND en consultation et vous addresse mes salutations confraternelles"). Une autre difficulté concerne les documents contenant plusieurs fois un même type de zone. Il s'agit par exemple de courriers avec des post-scriptum (plusieurs zones de contenu) ou de documents contenant le suivi d'un acte par plusieurs praticiens : les différents courriers échangés sont concaténés dans un seul document, qui comporte alors plusieurs zones de chaque type.

	Précision	Rappel	F-mesure
En-tête générique	1,0000	0,9868	0,9934
En-tête spécifique	0,9779	0,9705	0,9742
Contenu	0,9520	0,9806	0,9661
Pied de page	0,9187	0,7533	0,8278
Exactitude (accuracy)	**0,9678**		

TABLE 5 – Performances de la classification en zone des lignes des documents (corpus de test)

3.2 Performances de l'identification des zones sur le corpus de 500 documents

Les performances de la détection des débuts de zone sur le corpus de 500 documents sont présentées dans le tableau 6. Celles de la classification des lignes en zone sont données dans le tableau 7. On constate que les performances sont bonnes (F-mesure globale de 0,8875 pour la détection des débuts de zones et exactitude de 0,9550 pour la classification des lignes) et très similaires à celles obtenues sur le corpus de test.

	Précision	Rappel	F-mesure
Début d'en-tête générique	0,9843	0,9171	0,8875
Début d'en-tête spécifique	0,9391	0,8834	0,9104
Début de contenu	0,8834	0,7968	0,8379
Début de pied de page	0,8894	0,8446	0,8664
Micro-moyenne	**0,9210**	**0,8563**	**0,8875**

TABLE 6 – Performances de l'identification des débuts de zones (corpus de 500 documents)

	Précision	Rappel	F-mesure
En-tête générique	0,9860	0,9624	0,9741
En-tête spécifique	0,9750	0,9262	0,9500
Contenu	0,9161	0,9892	0,9513
Pied de page	0,9496	0,9009	0,9246
Exactitude (accuracy)	**0,9550**		

TABLE 7 – Performances de la classification en zone des lignes des documents (corpus de 500 documents)

3.3 Effet de l'identification du contenu principal sur la taille du corpus

Nous avons examiné la taille du corpus sélectionné (corpus de 500 documents), en calculant des stastistiques descriptives avant et après identification des zones de contenu : (1) des statistiques sur le corpus entier (avant identification des zones) et (2) des statistiques sur le contenu principal du corpus (après identification des zones). Les résultats sont présentés dans le tableau 8. La différence de taille entre le corpus brut et le corpus restreint aux zones de contenu est nette : un peu plus de 170 000 mots au total contre environ 100 000 mots de contenu. La longueur moyenne d'un document (en mots) est réduite de 41% (343 mots vs. 202 mots). En revanche, la longueur moyenne d'une phrase augmente de 22% par rapport au corpus brut. En effet, les en-têtes et pieds de pages contiennent davantage de phrases courtes par rapport au contenu principal qui comprend de longues phrases descriptives.

	Corpus brut	Zones de contenu	Différence (en %)
Nombre de mots	171 722	100 730	-41%
Nombre de phrases	18 815	9 013	-52%
Longueur moyenne d'un document (en mots)	343	202	-41%
Longueur moyenne d'un document (en phrases)	38	18	-53%
Longueur moyenne d'une phrase (en mots)	9	11	+22%

TABLE 8 – Statistiques descriptives sur le corpus de 500 documents

4 Discussion et conclusion

Les performances de notre méthode d'identification de zones sont très bonnes. En particulier, nous obtenons une précision de 0,9520 et un rappel de 0,9806 sur le corpus de test pour la classification des lignes de contenu. On constate donc que l'on perd très peu de contenu. Les performances semblent suffisamment hautes pour pouvoir baser la sélection d'un échantillon sur le contenu identifié automatiquement. Ceci est confirmé a posteriori par l'évaluation sur le corpus sélectionné de 500 documents (précision de 0,9161 et rappel de 0,9892 pour les lignes de contenu). Le corpus est sensiblement plus petit lorsque l'on se restreint au contenu principal (tableau 8). Ceci montre que les zones d'en-têtes et de pieds de pages sont très présentes dans les documents. Il est donc nécessaire de les identifier afin de travailler sur le contenu médical.

Remerciements

Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant la référence ANR-13-JS02-0009-01.

Références

DENNY J. C., SPICKARD III A., JOHNSON K. B., PETERSON N. B., PETERSON J. F. & MILLER R. A. (2009). Evaluation of a method to identify and categorize section headers in clinical documents. *Journal of the American Medical Informatics Association*, **16**(6), 806–815.

HIROHATA K., OKAZAKI N., ANANIADOU S. & ISHIZUKA M. (2008). Identifying sections in scientific abstracts using conditional random fields. In *Proceedings of the IJCNLP 2008*.

LAFFERTY J. D., MCCALLUM A. & PEREIRA F. C. N. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, p. 282–289, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

LAMPERT A., DALE R. & PARIS C. (2009). Segmenting email message text into zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, p. 919–928.

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.

LI Y., LIPSKY GORMAN S. & ELHADAD N. (2010). Section classification in clinical notes using supervised hidden markov model. In *Proceedings of the 1st ACM International Health Informatics Symposium*, p. 744–750 : ACM.

LIN J., KARAKOS D., DEMNER-FUSHMAN D. & KHUDANPUR S. (2006). Generative content models for structural analysis of medical abstracts. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, p. 65–72 : Association for Computational Linguistics.

MCKNIGHT L. & SRINIVASAN P. (2003). Categorization of sentence types in medical abstracts. In *AMIA Annual Symposium Proceedings*, volume 2003, p. 440 : American Medical Informatics Association.

MERITY S., MURPHY T. & CURRAN J. R. (2009). Accurate argumentative zoning with maximum entropy models. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, p. 19–26 : ACL.

RUCH P., GEISSBÜHLER A., GOBEILL J., LISACEK F., TBAHRITI I., VEUTHEY A. & ARONSON A. R. (2007). Using discourse analysis to improve text categorization in medline. In *Stud Health Technol Inform*, volume 129, p. 710–5.

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 Session at EACL 2012*, p. 102–7.

TEPPER M., CAPURRO D., XIA F., VANDERWENDE L. & YETISGEN-YILDIZ M. (2012). Statistical section segmentation in free-text clinical records. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

TEUFEL S. (2000). *Argumentative zoning : Information extraction from scientific text*. PhD thesis, Citeseer.

TEUFEL S. & MOENS M. (2002). Summarizing scientific articles : experiments with relevance and rhetorical status. *Computational linguistics*, **28**(4), 409–445.

YAMAMOTO Y. & TAKAGI T. (2005). A sentence classification system for multi biomedical literature summarization. In *Data Engineering Workshops, 2005. 21st International Conference on*, p. 1163–1163 : IEEE.

Un schéma d'annotation en dépendances syntaxiques profondes pour le français *

Guy Perrier [1] Marie Candito [2] Bruno Guillaume [3]
Corentin Ribeyre [2] Karën Fort [1] Djamé Seddah [4]
(1) Université de Lorraine/LORIA (2) Université Paris Diderot/INRIA
(3) Inria Nancy Grand-Est/LORIA (4) Université Paris Sorbonne/INRIA

Résumé. À partir du schéma d'annotation en dépendances syntaxiques de surface du corpus Sequoia, nous proposons un schéma en dépendances syntaxiques profondes qui en est une abstraction exprimant les relations grammaticales entre mots sémantiquement pleins. Quand ces relations grammaticales sont partie prenante de diathèses verbales, ces diathèses sont vues comme le résultat de redistributions à partir d'une diathèse canonique et c'est cette dernière qui est retenue dans notre schéma d'annotation syntaxique profonde.

Abstract. We describe in this article an annotation scheme for deep dependency syntax, built from the surface annotation scheme of the Sequoia corpus, abstracting away from it and expressing the grammatical relations between content words. When these grammatical relations take part into verbal diatheses, we consider the diatheses as resulting from redistributions from the canonical diathesis, which we retain in our annotation scheme.

Mots-clés : schéma d'annotation, syntaxe profonde, grammaires de dépendance.

Keywords: annotation scheme, deep syntax, dependency grammar.

1 Introduction

Les corpus annotés en dépendances syntaxiques présentent un intérêt croissant par rapport aux corpus annotés en syntagmes dans la mesure où ils permettent plus directement d'extraire les relations prédicat-argument constitutives d'une représentation sémantique. Cette extraction reste cependant non triviale, la syntaxe offrant une grande variabilité dans la façon d'exprimer ces relations. Une solution consiste à définir un niveau intermédiaire, la syntaxe profonde, qui est une abstraction de la syntaxe de surface et qui vise à se rapprocher du niveau sémantique.

La Théorie Sens-Texte (TST)(Mel'čuk, 1988), parmi les différents niveaux de la langue qu'elle définit, propose un niveau de syntaxe profonde, où seuls les mots sémantiquement pleins sont présents. Ce niveau de syntaxe profonde est lié à la sémantique dans la mesure où les mots présents renvoient aux entrées d'un lexique sémantique.

Comme le pointe Zabokrtsky (2005), la TST est très proche du modèle de la Description Générative Fonctionnelle ou Functional Generative Description (FGD) (Sgall *et al.*, 1986), sur lequel s'appuie l'annotation du Prague Dependency Treebank (Hajic *et al.*, 2006). La FGD définit un niveau tectogrammatical qui s'apparente au niveau de syntaxe profonde de la TST avec une différence dans son rapport à la sémantique : l'absence de lien avec un lexique sémantique est compensée par un étiquetage des dépendances prédicat-argument avec des rôles thématiques.

Contrairement à ces deux modèles, le schéma de syntaxe profonde que nous proposons ne marque aucun engagement sémantique, la sémantique ne servant qu'à lever les ambiguïtés de rattachement. Nous ne retenons que les mots sémantiquement pleins mais les relations entre ceux-ci expriment des fonctions grammaticales. Les diathèses verbales présentes en surface sont considérées comme le résultat de redistributions à partir de diathèses canoniques et ce sont ces diathèses canoniques qui sont représentées au niveau profond.

Nous sommes partis de l'annotation de surface du corpus Sequoia (Candito & Seddah, 2012b) et nous avons réalisé son annotation en syntaxe profonde, en nous inspirant des méthodes de Bonfante *et al.* (2011). Le corpus Sequoia annoté

*. Ce travail a été partiellement financé par l'Agence Nationale de la Recherche au titre du programme Investissements d'Avenir, au sein du Labex EFL (ANR-10-LABX-0083). Nous avons bénéficié de discussions avec Alain Polguère et Sylvain Kahane, que nous remercions.

en syntaxe profonde est librement disponible [1]. Si nous décrivons, dans (Candito *et al.*, 2014), l'ensemble du travail d'annotation du corpus, nous nous concentrons ici sur la présentation du schéma d'annotation. Une description plus précise des phénomènes est fournie dans le guide d'annotation sur le site du corpus.

2 Schéma d'annotation en syntaxe profonde

Nous avons défini notre schéma d'annotation en syntaxe profonde en partant du schéma d'annotation en dépendances de surface du corpus Sequoia, ce qui a eu une influence certaine sur le résultat. Il est donc important de détailler comment le schéma de surface a été lui-même défini : le corpus Sequoia a été annoté d'abord en arbres de constituants, en suivant le schéma d'annotation du corpus arboré de Paris 7 (Abeillé & Barrier, 2004) ou French Treebank (FTB). Le corpus en constituants a ensuite été converti automatiquement en dépendances de surface, en suivant la procédure décrite dans (Candito *et al.*, 2010). Les arbres de dépendances résultants suivent ainsi très largement les choix linguistiques du FTB (Abeillé *et al.*, 2004; Abeillé, 2004), dans la mesure où la majorité des phénomènes syntaxiques sont mécaniquement traduits en dépendances [2]. Enfin, les dépendances longue distance ont été corrigées manuellement (Candito & Seddah, 2012a) dans les arbres de dépendance obtenus par conversion automatique, ce qui a introduit quelques arcs non projectifs [3].

Le schéma d'annotation de surface résultant est notre point de départ. Nous avons cherché, pour des raisons pragmatiques, à minimiser les divergences entre les niveaux surfacique et profond, pour nous concentrer sur les phénomènes non directement représentables dans les arbres de surface. Aussi, par exemple, avons-nous conservé la réprésentation des coordinations avec le premier conjoint comme tête, bien que cela ne permette pas de distinguer les dépendants partagés par plusieurs conjoints des dépendants du seul premier conjoint.

2.1 Choix théoriques

L'objectif principal de nos représentations syntaxiques profondes (REPRSYNTPROF dans la suite) est de généraliser sur la variation syntaxique autant que possible sans faire de distinctions ni de généralisations purement sémantiques. Nous utilisons pour cela la notion de *sous-catégorisation canonique* et représentons les changements de diathèse comme des redistributions des fonctions grammaticales sous-catégorisées par un lexème. Nous inspirant de la Grammaire Relationnelle (Perlmutter, 1983), nous distinguons fonction grammaticale *canonique* et fonction grammaticale *finale* d'une part, et *cadre de sous-catégorisation canonique* (CS canonique) et *cadre de sous-catégorisation final* (CS final) [4].

Définissons d'abord la notion de sous-catégorisation finale pour un verbe : elle contient d'une part les fonctions finales associées aux arguments *exprimés* du verbe, et d'autre part, dans le cas d'ellipse et/ou de verbes non conjugués, les fonctions des éléments qui seraient des arguments du verbe si celui-ci était utilisé sans ellipse et conjugué. Cette formulation permet de faire entrer dans le CS final par exemple le sujet des infinitifs, le sujet ou l'objet des participes épithètes, le sujet de verbes coordonnés ou plus généralement tout argument partagé par plusieurs prédicats. Par exemple, dans *« Anna veut dormir, mais devra peut-être veiller »*, le CS final de *dormir* est [sujet] rempli par *Anna*, et le CS final de *devra* est [sujet, objet], rempli par *Anna* et *veiller*. Entrent également dans le CS final l'élément modifié par un participe épithète : par exemple pour *« les personnes nées en 40 »*, le nom *personnes* est la tête du sujet final de *nées*.

Passons maintenant à la définition précise de la sous-catégorisation canonique. Afin de neutraliser la variation syntaxique due aux changements de diathèse, nous considérons ceux-ci comme des redistributions des fonction canoniques associées aux arguments syntaxiques. Suivant la Grammaire Relationnelle (Perlmutter, 1983), le CS final est vu comme résultant de l'application de 0 à n redistributions sur un CS canonique. Etant donnée une occurrence de verbe, le CS canonique peut donc par définition être obtenu par application inverse des redistributions appropriées. Un exemple simple est le cas d'un verbe au passif dont le CS final est [sujet, par-objet] et le CS canonique est [sujet, objet]. Les éléments du CS canonique, appelés *arguments syntaxiques profonds*, sont obligatoirement sémantiquement pleins. Le *il* explétif n'appartient qu'au CS final. Donc dans l'exemple *« Trois personnes arrivent »*, à partir du CS canonique [sujet], la redistribution de l'impersonnel rétrograde le sujet en objet direct, et un *il* explétif remplit la fonction sujet final : *« Il arrive trois personnes »* a pour CS canonique [sujet] et pour CS final [sujet, objet].

Pour définir nos REPRSYNTPROF, nous n'avons considéré que les redistributions qui comportent un marquage morpho-syntaxique (typiquement l'auxiliaire pour le passif, ou le clitique sémantiquement vide *se* pour les alternances moyennes

1. http ://deep-sequoia.inria.fr
2. Des informations supplémentaires sont prédites par la procédure de conversion, en cas de sous-spécification dans la version en constituants (c'est le cas pour les étiquettes de dépendances pour les dépendants de gouverneurs non verbaux).
3. Par exemple, à la conversion de l'arbre syntagmatique de *« ... le succès que la municipalité était en droit d'attendre »*, le pronom relatif est mécaniquement rattaché comme dépendant de *était*, et manuellement corrigé pour dépendre de *attendre*.
4. La Grammaire Relationnelle utilise l'adjectif *initial* plutôt que *canonique*.

et neutres). Les alternances syntaxiques sans marquage morpho-syntaxique ne sont pas capturées au sein de nos REPRSYNTPROF, et donnent lieu à des CS canoniques différents. Les repérer, en l'absence de marquage formel, relève pour nous de l'analyse sémantique. Par exemple, pour le verbe *baisser*, les deux emplois *X baisse Y* et *Y baisse*, typiquement reliés par l'alternance causative/inchoative non marquée, ne sont pas reliés dans nos REPRSYNTPROF, et ont deux CS canoniques distinctes [sujet, objet] et [sujet] respectivement, qui sont aussi ici les CS finaux en l'absence de redistribution. En revanche, pour l'alternance moyenne (par exemple *« On avale facilement ce médicament / Ce médicament s'avale facilement »*) ou l'alternance neutre (qui "efface" l'actant agentif ou causal, comme dans *« Cela dissout le médicament / Le médicament se dissout (de lui-même) »*), le lien entre les deux versions est capturé par redistribution, et pour ces deux alternances, l'objet direct dans la version transitive (*médicament*) est sujet final mais objet canonique dans la version intransitive. Nous retenons comme redistributions : le passif, l'impersonnel, le moyen, le neutre et le causatif (voir les exemples en section 3)[5], certaines pouvant interagir. Nous renvoyons à (Candito, 1999) pour une étude des interactions entre redistributions pour le français.

Nous avons volontairement fait deux distinctions différentes : représentation profonde *versus* de surface, et fonction grammaticale finale *versus* canonique. Cela nous est utile pour capturer par exemple la régularité concernant le contrôle des sujets des infinitifs : il s'agit du sujet *final* de l'infinitif, quelle que soit la diathèse de celui-ci. Ainsi, pour *« Paul veut être embauché »*, dans la représentation de surface, *Paul* est le sujet final de *veut*. Dans la représentation profonde, il est aussi le sujet *final* de *être embauché* et son objet canonique.

Outre les verbes, nous traitons dans nos annotations les adjectifs, auxquels la notion de sous-catégorisation peut être étendue, mais sans distinction entre final et canonique (ce que nous ne détaillons pas ici par manque de place). Le travail reste à faire pour les autres catégories de prédicats, en particulier les noms.

Nous pouvons maintenant lister précisément les informations qui sont explicitées dans nos REPRSYNTPROF par rapport aux représentations de surface :
– le statut sémantique des mots de la phrase (sémantiquement vide ou plein),
– la liste complète des arguments syntaxiques profonds des verbes et des adjectifs,
– et leur fonction grammaticale canonique (et la ou les redistributions amenant à l'emploi observé en surface).
Dans nos REPRSYNTPROF, chaque argument syntaxique profond d'un prédicat lui est directement rattaché. Plus précisément, nous prenons comme tête d'un argument syntaxique profond l'élément sémantiquement plein le plus haut (les complémenteurs vides et les prépositions régies sont court-circuitées, comme le sont aussi les pronoms relatifs dont la référence peut être résolue syntaxiquement, comme illustré dans la phrase (2) de la figure 1).

2.2 Caractéristiques formelles

Nous définissons une représentation *complète* comme un graphe de dépendances contenant à la fois la REPRSYNTSURF et la REPRSYNTPROF. La figure 1 fournit des exemples, tirés du corpus Sequoia, de représentations complètes. Les nœuds sont les mots de la phrase (ou des composants de composés réguliers), et sont typés comme sémantiquement vides (en rouge dans la figure) ou pleins (en noir). Les arcs portent :
– une information sur leur appartenance à la REPRSYNTSURF ou pas, et leur appartenance à la REPRSYNTPROF ou pas : un arc peut être surfacique mais non profond (arcs rouges), profond mais non surfacique (arcs bleus), et à la fois profond et surfacique (arcs noirs) ;
– une étiquette qui est constituée soit d'une seule fonction, à la fois finale et canonique, pour les fonctions n'intervenant jamais dans des changements de diathèse (comme MOD), soit de la fonction finale et de la fonction canonique (dans toute la suite, une étiquette notée "$f : c$" correspond à la fonction finale f et la fonction canonique c).
La REPRSYNTPROF pour une phrase donnée est formée des nœuds sémantiquement pleins et des arcs profonds, c'est-à-dire à la fois les arcs qui sont profonds et surfaciques (noirs) et les arcs profonds non surfaciques (bleus). Il s'agit formellement d'un graphe orienté, qui peut contenir des cycles et des arcs multiples (un même couple gouverneur / dépendant peut être relié par plusieurs arcs de même orientation mais d'étiquettes différentes).

3 Illustration sur des exemples de phénomènes relatifs au verbe

Dans cette section, nous montrons comment le schéma qui vient d'être défini s'applique à certains phénomènes linguistiques intéressants. Compte tenu des contraintes de l'article, nous nous limitons à des phénomènes relatifs au verbe. Les exemples présentés dans la figure 1 illustrent ces phénomènes et les interactions complexes qui existent entre l'ajout de relations prédicat-argument liées à la syntaxe profonde, la redistribution entre les fonctions grammaticales canoniques et

5. Le réfléchi, bien que présentant des propriétés *« intransitivantes »* qui peuvent justifier une représentation *via* redistributions, n'est pas traité dans nos REPRSYNTPROF via redistribution (cf. l'exemple (1), dans la figure 1).

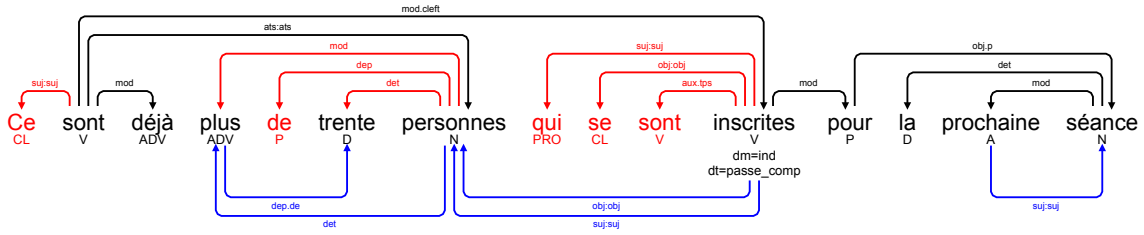
finales, et la suppression de mots sémantiquement vides. [6]

Transformation de marqueurs grammaticaux en traits Sont concernés les auxiliaires et les clitiques ne représentant aucun argument mais destinés à modifier la sémantique du verbe auquel ils s'appliquent.

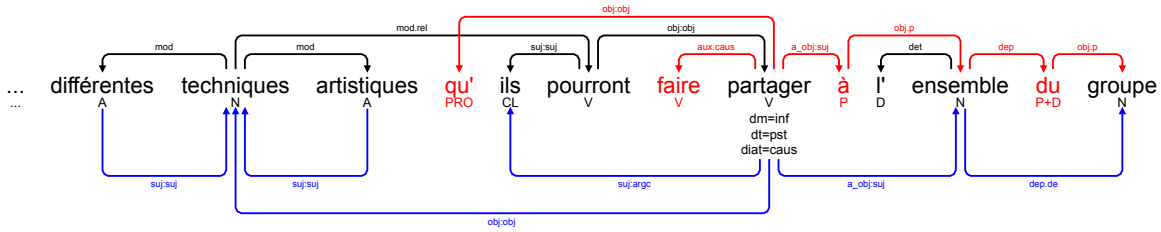
En REPRSYNTSURF, les auxiliaires (pour les temps composés, les passifs et les causatifs) sont systématiquement rattachés au dernier verbe du noyau verbal (le verbe sémantiquement plein) et ils sont supprimés en REPRSYNTPROF. L'information de mode, de temps et éventuellement de diathèse dont ils sont porteurs est conservée sous forme de traits profonds attachés au verbe plein.

Les clitiques réfléchis intrinsèques *se* sont également traités comme des marqueurs et systématiquement transformés en traits (cf. infra la section spécifique sur les réfléchis).

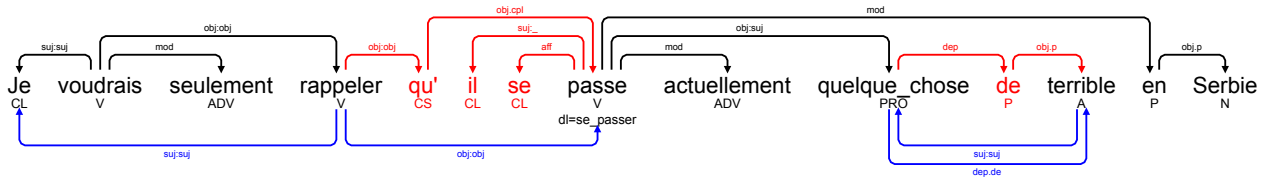
(1) auxiliaire, réfléchi, clivée



(2) causatif, verbe à montée, relative épithète



(3) réfléchi, impersonnel



(4) réfléchi, coordination

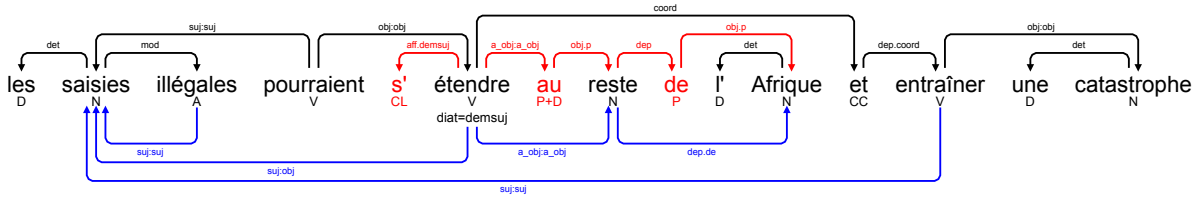


FIGURE 1 – Exemples d'annotation de certains phénomènes. La REPRSYNTPROF est formée des mots en noir, des arcs noirs et des arcs bleus. La REPRSYNTSURF contient tous les mots, et les arcs rouges et noirs.

Les causatifs La phrase (2) illustre une construction causative dont le noyau est *« faire partager »* et le passage de la REPRSYNTSURF à la REPRSYNTPROF montre l'interférence entre plusieurs phénomènes.

Le verbe *pourront* est ici un verbe à montée du sujet, si bien qu'*ils* est sujet profond final de *partager*. Le changement de la diathèse finale causative en diathèse canonique fait qu'*ils* passe de la fonction finale SUJ à la fonction canonique

6. Les schémas sont simplifiés et ne font apparaître que les traits pertinents concernant les phénomènes décrits dans l'article. Dans la version finale, une adresse URL renverra à un tableau complet des traits utilisés.

ARGC, pour *« argument causateur »*. Parallèlement, *à* passe de AOBJ à SUJ, mais comme en REPRSYNTPROF, la préposition régie est effacée, ces fonctions sont transférées sur *ensemble*.

Enfin, l'auxiliaire causatif *faire* est supprimé et une indication de diathèse causative (`diat=caus`) est ajoutée au verbe *partager* [7].

Les clitiques réfléchis Le clitique réfléchi *se* a été annoté de trois manières différentes. Dans la phrase (1), *se* est un vrai réfléchi, objet direct du verbe *inscrites* en REPRSYNTSURF et il disparaît en REPRSYNTPROF ; sa fonction OBJ est alors reportée sur son antécédent *personnes* (via le pronom relatif *qui* qui disparaît lui aussi en REPRSYNTPROF). Dans la phrase (3), *se* est un réfléchi intrinsèque. Affixe du verbe essentiellement pronominal *se passer*, il disparaît en REPRSYNTPROF, où il est remplacé par un trait de lemme profond `dl=se_passer` attaché à *passer*. Enfin, dans la phrase (4), *s'* avec le verbe *étendre* s'inscrit dans une diathèse neutre. Cette diathèse provient d'une redistribution dans laquelle l'objet canonique devient le sujet final et le sujet canonique n'a aucun référent clairement identifié. Cela la distingue de la diathèse moyenne où ce référent n'est pas exprimé mais existe. Comme il est difficile de distinguer les deux constructions, nous les marquons toutes deux en REPRSYNTPROF par un trait `diat=demsuj` attaché au verbe, *étendre* dans notre exemple.

Les sujets impersonnels Le clitique *il* explétif peut apparaître soit comme sujet final de verbes essentiellement impersonnels, soit dans le cas de diathèses impersonnelles de verbes qui peuvent également apparaître avec des sujets référentiels. La phrase (3), avec *« il se passe »*, illustre le second cas. Le clitique *il* disparaît en REPRSYNTPROF et l'objet final *quelque chose* y est sujet canonique.

Les coordinations Selon le schéma d'annotation du FTB, la tête de la structure de coordination est la tête du premier conjoint. La conjonction de coordination est reliée à la tête avec une étiquette COORD et la tête du second conjoint à la conjonction avec une étiquette DEP.COORD. Ces dépendances sont présentes aussi bien en surface qu'en profondeur. Pour des raisons sémantiques, nous traitons au niveau profond les dépendances entrantes différemment des dépendances sortantes. D'un point du vue entrant, une coordination est vue comme un tout et les dépendances ne sont pas distribuées entre les conjoints (autrement dit, nous ne désambiguïsons pas entre lecture collective et distributive de la coordination). En revanche, les dépendants d'un élément coordonné qui sont reliés à son premier conjoint en surface sont distribués aux autres conjoints en profondeur (sauf exception non détaillée ici). Ainsi, dans la phrase (4), le sujet profond *saisies* des conjoints *étendre* et *entraîner* est distribué, alors que le gouverneur de la coordination *pourraient* n'est pas distribué.

Les clivées La phrase (1) est une clivée : le syntagme nominal *« plus de trente personnes »* est extrait de la phrase canonique *« plus de trente personnes se sont inscrites pour la prochaine séance »* pour être placé en attribut du sujet de *sont*, afin que l'accent soit mis sur lui. La trace du syntagme extrait est représentée par le pronom relatif *qui* et la phrase canonique devient une proposition relative rattachée à *sont* par une dépendance MOD.CLEFT.

En REPRSYNTPROF, la phrase canonique n'est pas complètement restaurée et la dépendance MOD.CLEFT est conservée afin de permettre au verbe *sont* de recevoir des modificateurs, comme l'adverbe *déjà* dans l'exemple. Seuls *ce* et *qui* sont supprimés et la fonction *sujet* du pronom relatif est transférée sur son antécédent *personnes*.

4 Représentation syntaxique profonde versus représentation sémantique

Si nos REPRSYNTPROF partagent certaines caractéristiques avec des représentations sémantiques (qui sont parfois appelées *profondes*), elles s'en différencient sur des points importants. Le premier point est que dans nos REPRSYNTPROF, le sens des nœuds lexicaux n'est pas désambiguisé. La sémantique n'est utilisée que pour désambiguiser des rattachements syntaxiques ou des changements de diathèse. Ensuite, même si les mots sémantiquement vides sont écartés de nos REPRSYNTPROF, les nœuds restants ne forment pas nécessairement une unité sémantique. En effet, les unités polylexicales (expressions figées, constructions à verbe support, composés syntaxiquement réguliers) ne sont pas marquées comme formant une unité, et sont représentées par une structure syntaxique régulière.
En outre, la sous-catégorisation canonique des prédicats explicitée dans les REPRSYNTPROF peut contenir des éléments qui ne sont pas des arguments sémantiques du prédicat (comme par exemple le sujet *ils* du verbe à montée *pourront* de la phrase (2), dans la figure 1). Enfin, alors que les représentations sémantiques typent en général les actants sémantiques des prédicats au moyen de rôles sémantiques ou de simples numéros, nos REPRSYNTPROF utilisent seulement les fonctions canoniques, ce qui est cohérent avec le fait que les lemmes ne sont pas désambiguisés. Les fonctions demeurent alors des indices importants pour la (future) désambiguisation des prédicats.

Pour résumer, pour obtenir des structures prédicat-arguments (sémantiques) à partir de nos REPRSYNTPROF, il serait nécessaire de repérer les unités polylexicales, désambiguiser les prédicats, ne retenir que les arguments sémantiques de

7. Par manque de place, nous ne détaillons pas ici les différents types de causatifs.

ceux-ci parmi les arguments syntaxiques, et les associer à un rôle sémantique ou une simple numérotation. Des numéros d'argument peuvent être obtenus en utilisant un ordre d'oblicité des fonctions grammaticales canoniques (typiquement sujet < objet < obliques). A noter cependant que les sujets canoniques de nos REPRSYNTPROF peuvent correspondre aussi bien à des proto-agents qu'à des proto-patients (au sens de (Dowty, 1991)), cette distinction étant considérée comme sémantique. Nos REPRSYNTPROF ne capturent donc pas que dans une alternance causative/inchoative, par exemple *X baisse Y* versus *Y baisse*, l'argument *Y* joue le même rôle sémantique. Cette généralisation est capturée notamment dans la ressource PropBank (Palmer *et al.*, 2005) en utilisant un même *rôle* arg1 pour l'argument *Y* dans les deux variantes. D'autres travaux prévoient plutôt un lien au niveau du lexique, entre le lexème transitif et le lexème intransitif : c'est ce qui est prévu dans le dictionnaire explicatif et combinatoire de la TST (Mel'čuk *et al.*, 1999), et dans le DeepBank (Flickinger *et al.*, 2012), constitué de représentations syntactico-sémantiques dérivées d'analyses HPSG de l'anglais (Oepen, p.c). Quoi qu'il en soit, nous espérons que la libre mise à disposition du corpus et de son guide d'annotation permettront tant le développement de ressources syntaxico-sémantiques plus riches que des perspectives nouvelles d'application.

References

ABEILLÉ A. (2004). Guide des annotateurs, annotation fonctionnelle. *LLF Annotation Guide*.

ABEILLÉ A. & BARRIER N. (2004). Enriching a French treebank. In *Proc. of LREC*, Lisbonne, Portugal.

ABEILLÉ A., TOUSSENEL F. & MARTINE C. (2004). Corpus le monde, annotations en constituants, guide pour les correcteurs. *LLF Annotation Guide*.

BONFANTE G., GUILLAUME B., MOREY M. & PERRIER G. (2011). Enrichissement de structures en dépendances par réécriture de graphes. In *Proc. of TALN*, Montpellier, France.

CANDITO M. (1999). *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées application au français et à l'italien*. PhD thesis, Université Paris Diderot.

CANDITO M., CRABBÉ B. & DENIS P. (2010). Statistical french dependency parsing: Treebank conversion and first results. In *Proc. of LREC*, La Valette, Malte.

CANDITO M., PERRIER G., GUILLAUME B., RIBEYRE C., FORT K., SEDDAH D. & DE LA CLERGERIE É. (2014). Deep Syntax Annotation of the Sequoia French Treebank. In *International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Islande.

CANDITO M. & SEDDAH D. (2012a). Effectively long-distance dependencies in French: annotation and parsing evaluation. In *Proc. of TLT 11*, Lisbonne, Portugal.

CANDITO M. & SEDDAH D. (2012b). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Proc. of TALN*, Grenoble, France.

DOWTY D. (1991). Thematic proto-roles and argument selection. *Language*, **67**(3), 547–619.

FLICKINGER D., ZHANG Y. & KORDONI V. (2012). Deepbank: A dynamically annotated treebank of the wall street journal. In *Proc. of the Eleventh International Workshop on Treebanks and Linguistic Theories*, p. 85–96.

HAJIC J., PANEVOVÁ J., HAJICOVÁ E., SGALL P., PAJAS P., ŠTEPÁNEK J., HAVELKA J., MIKULOVÁ M., ZABOKRTSKÝ Z. & RAZIMOVÁ M. Š. (2006). Prague dependency treebank 2.0. *CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphie*, **98**.

MEL'ČUK I. (1988). *Dependency syntax: theory and practice*. State University Press of New York.

MEL'ČUK I., ARBATCHEWSKY-JUMARIE N., IORDANSKAJA L., MANTHA S. & POLGUÈRE A. (1999). *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques IV* [Explanatory and Combinatorial Dictionary of Contemporary French. Lexico-Semantic Research IV]. Montréal: Les Presses de l'Université de Montréal.

PALMER M., GILDEA D. & KINGSBURY P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, **31**(1), 71–106.

PERLMUTTER D. (1983). *Studies in Relational Grammar 1*. University of Chicago Press.

SGALL P., HAJIČOVÁ E. & PANEVOVÁ J. (1986). *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Dordrecht:Reidel Publishing Company and Prague:Academia.

ZABOKRTSKY Z. (2005). Resemblances between Meaning-Text Theory and Functional Generative Description. In *Proc. of MTT 2005*, p. 549–557, Moscou, Russie.

Analyse argumentative du corpus de l'ACL (ACL Anthology)

Untel Trucmuche[1, 2] Unetelle Machinchose[1, 3]
(1) LPL, AMU, CNRS, 5 avenue Pasteur, 13100 Aix-en-Provence
(2) LIF, AMU, CNRS, 163 avenue de Luminy, 13288 Marseille Cedex 9
(3) Lab, adresse, CP Ville, Pays
utrucmuche@lpl-aix.fr, umachinchose@adresse-academique.fr

Résumé. Cet article présente un essai d'application de l'analyse argumentative (*text zoning*) à l'ACL Anthology. Il s'agit ainsi de mieux caractériser le contenu des articles du domaine de la linguistique informatique afin de pouvoir en faire une analyse fine par la suite. Nous montrons que des technique récentes d'analyse argumentative fondées sur l'apprentissage faiblement supervisé permettent d'obtenir de bons résultats.

Abstract. This paper presents an application of Text Zoning to the ACL Anthology. Text Zoning is known to be useful to characterize the content of papers, especially in the scientific domain. We show that recent techniques based on weakly supervised learning obtain excellent results on the ACL Anthology. Although these kinds of techniques is known in the domain, it is the first time it is applied to the whole ACL Anthology.

Mots-clés : Analyse argumentative, corpus de textes scientifiques, ACL Anthology.

Keywords: Text Zoning, Corpus of scientific texts, ACL Anthology.

1 Introduction

L'analyse des masses de données (en anglais *big data*) est un thème de recherche porteur aujourd'hui. Les masses de données permettent en effet de mettre au jour des phénomènes difficilement observables sans méthodes automatiques, et la numérisation de tous les secteurs de la société permet aujourd'hui d'avoir accès à des données en grandes quantités pour un grand nombre de domaines.

La science est un des domaines qui produit ainsi de nombreuses données informatisées (littérature scientifique, mais aussi données brutes sous forme de textes, d'images, de chiffres, etc.) et la numérisation des données passées permet aujourd'hui d'avoir accès, pour plusieurs domaines très variées, à des collections d'articles de recherche s'étendant sur plusieurs dizaines d'années. Le monde de la linguistique informatique n'est pas en reste et l'ACL Anthology met aujourd'hui à la disposition des chercheurs plus de 24500 articles au format PDF. Les plus anciens articles datent de 1965 (première édition de COLING) mais ce n'est qu'à partir des années 1980 qu'on commence à avoir des données relativement conséquentes, le volume allant grandissant chaque année depuis lors (il y a donc une très grande disparité dans les volumes de données disponibles suivant les années considérées). Il existe des bases de données similaires pour la biologie et le domaine biomédical (par ex. Medline), les systèmes complexes ou la physique (par ex. APS data set de la *American Physical Society*) pour citer quelques bases ayant fait l'objet d'enquêtes diverses.

L'ACL Anthology a reçu un intérêt particulier en 2012 pour les 50 ans de l'*Association for Computational Linguistics*. Un atelier s'intitulant "*Rediscovering 50 Years of Discoveries*" a été organisé cette année-là (Banchs, 2012) : il s'agissait pour l'association de jeter un regard sur l'évolution du domaine depuis 50 ans. Au-delà de ces circonstances particulières, cet événement a été l'occasion d'analyser les données accumulées depuis 50 ans (mais pour les raisons données plus haut, la plupart des études portent sur les articles produits depuis 1980) avec les outils modernes issus à la fois du traitement des langues et des systèmes complexes, afin d'analyser l'évolution du domaine. Ce type d'analyse reste toutefois superficiel si une analyse fine de la structure des articles ou de leur résumé n'est pas effectuée au préalable.

La reconnaissance et l'annotation de la structure discursive des articles scientifiques présentent en effet des enjeux importants pour la communauté scientifique mais aussi pour le monde de l'édition. Ce type de techniques peut en effet permettre

de savoir si une section d'un article scientifique donné concerne, par exemple, le protocole expérimental employé, les données d'expériences ou la discussion et la comparaison avec les travaux antérieurs. Ce type d'analyse donne des résultats de plus en plus précis et commence à intéresser les grandes maisons d'éditions scientifiques, dans la mesure où on peut ainsi enrichir les bases de connaissances existantes et proposer de nouveaux parcours de lecture.

Nous présentons dans cet article une application de la reconnaissance automatique de la structure argumentative à l'ensemble des articles du corpus de l'ACL (*ACL Anthology*). Notre but à terme est d'analyser ce corpus suivant différents plans : évolution des thèmes de recherche au cours du temps, positionnement des acteurs dans le domaine, évolution des techniques utilisées, etc. Il est du coup primordial de pouvoir caractériser la structure des articles et/ou de leur résumé : les citations, les mentions des techniques employées ou des méthodes d'évaluation sont très dépendantes de leur contexte d'emploi comme l'ont démontré les travaux de S. Teufel depuis une quinzaine d'années. Il est par exemple essentiel de savoir si une technique d'analyse est citée dans les travaux antérieurs ou est réellement celle utilisée par l'auteur de l'article.

Nous présentons un rapide état de l'art présentant les évolutions récentes de l'annotation argumentative. Nous présentons ensuite l'application de ce type de techniques à l'ACL Anthology. La section suivante est consacrée aux résultats et à la discussion, avant la conclusion.

2 Etat de l'art

Les premiers travaux d'importance en matière d'analyse de la structure rhétorique sont certainement ceux de Simone Teufel (Teufel, 1999) qui a proposé de catégoriser les phrases d'articles de traitement automatique des langues suivant sept étiquettes différentes : BKG (arrière-plan scientifique), OTH (description neutre de travaux antérieurs), OWN (description neutre du travail de l'auteur), AIM (objectifs de l'article), TXT (annonce de l'organisation de l'article), CTR (comparaison avec des travaux antérieurs) et BAS (description des travaux antérieurs sur lesquels s'appuie l'article).

La tâche est appelée « rhetorical zoning » ou « argumentative zoning » par l'auteur, dans la mesure où le balisage doit permettre d'identifier la fonction rhétorique ou argumentative de chaque phrase du texte.

Le travail initial de S. Teufel (Teufel, 1999) est fondé sur l'annotation manuelle de 200 articles représentatifs du domaine issus des conférences de l'ACL et de la revue *Computational Linguistics*. Un classifieur est ensuite entraîné sur cette base : il permet d'obtenir une annotation automatique de nouveaux textes donnés en entrée à l'analyseur. L'auteur rapporte que le système automatique donne le bon résultat dans 70% des cas, comparé à un accord de 88% entre humains. Le classifieur repose sur un modèle bayesien naïf car les méthodes plus sophistiquées testées par l'auteur ne semblent pas permettre d'obtenir de meilleurs résultats.

Teufel montre dans une publication ultérieure (Teufel & Moens, 2002) comment cette technique peut être utilisée pour générer des résumés automatiques de qualité. Les techniques de résumé traditionnelles sont fondées sur la sélection de phrases en fonction de leur intérêt informatif supposé, essentiellement sur la base des noms et des verbes qui la compose (les mots les plus centraux, souvent appelés centroïdes (Radev *et al.*, 2004)), ce qui pose problème pour générer des textes tenant compte de la variété du texte de départ. Le repérage de la structure argumentative répond partiellement à ce problème dans la mesure où il est dès lors possible de générer des résumés reflétant les différentes zones repérées ou, au contraire, privilégiant une zone donnée suivant les besoins informationnels du lecteur.

Teufel a enfin montré (Teufel *et al.*, 2006) comment le marquage argumentatif peut être couplé avec les références scientifiques. Les articles scientifiques sont en effet fondées sur des citations des travaux antérieurs mais ces citations peuvent avoir différentes finalités : simple mention de travaux antérieurs donnant l'arrière-plan de la recherche en cours, travaux précis auxquels s'oppose la publication en cours, référence à des travaux utilisant le même protocole expérimental, etc. Coupler repérage de référence et balisage argumentatif permet de typer les citations, toujours dans le but de faciliter la lecture en fonction des besoins informationnels du lecteur.

Les travaux de S. Teufel ont depuis donné lieu à différents types de travaux, d'une part pour affiner la méthode d'annotation, d'autre part pour vérifier son applicabilité à différents domaines scientifiques. Pour le premier point, les recherches ont porté sur les traits pertinents pour la classification, l'évaluation de différents algorithmes pour la tâche et surtout la diminution de la quantité de texte à annoter pour obtenir un système fonctionnel. Pour le second, c'est surtout le domaine de la biomédecine et de la biologie qui ont montré le plus d'intérêt pour ce type de techniques, du fait de la quantité d'articles disponible dans ce domaine et de la nécessité d'accéder de manière transversale à cette littérature (les biologistes peuvent par exemple avoir besoin d'accéder à tous les protocoles expérimentaux pour un problème donné) (Mizuta *et al.*,

2006; Tbahriti *et al.*, 2006).

Les travaux de Y. Guo (Guo *et al.*, 2011, 2013) reprennent l'analyse de la structure argumentative en complétant les travaux initiaux de S. Teufel sur un certain nombre de points : recours à une vaste liste de critères pour déterminer la classification des phrases, évaluation de plusieurs algorithmes d'apprentissage et diminution de la quantité de données annotées à fournir au système pour l'entraînement.

Y. Guo *et al.* (2011) proposent en particulier d'avoir recours à l'apprentissage peu supervisé pour entraîner leur système. On sait en effet que ce type d'approche permet de réduire la quantité de données annotées en utilisant parallèlement une grande masse de données non annotées : cette méthode est bien indiqué dans notre cas dans la mesure où les corpus à analyser en traitement des langues (et particulièrement l'ACL Anthology) sont souvent d'assez grande taille mais qu'il ne sont évidemment pas annotés. Les traits utilisés pour l'apprentissage sont de trois types : *i*) positionnels (localisation de la phrase au sein de l'article), *ii*) lexicaux (mots, classes de mots, bigrammes, etc. sont pris en considération) et *iii*) syntaxiques (les différentes relations syntaxiques, ainsi que les classes de noms en position sujet et les classes de noms en position objet sont pris en considération). L'analyse est donc considérablement plus riche que celle de Teufel mais nécessite en contrepartie un analyseur syntaxique.

Au niveau du modèle d'apprentissage, Guo et al. (2011) propose une méthode originale fondée sur l'apprentissage actif. Pour l'apprentissage, les SVM actifs (*Active SVM*) se fondent sur un nombre limité d'exemples annotés mais sont capables de repérer des configurations où l'algorithme risque de donner des résultats incertains. Ces configurations particulières doivent alors faire l'objet d'une annotation manuelle, ce qui permet à l'algorithme de progresser rapidement en se focalisant uniquement sur les cas limites en matière de catégorisation. Le repérage des données à annoter est souvent fondé sur la structure de l'apprentissage au moyen de SVM : par exemple, les points qui sont à la frontière de l'hyperplan permettant la catégorisation sont souvent pris en considération en priorité (Tong & Koller, 2001; Novak *et al.*, 2006).

Dans le cadre de notre expérience, les configurations particulières faisant l'objet d'un étiquetage manuel sont ceux pour lesquels l'algorithme ne distingue pas de classe majoritaire de manière claire. Ce sont les phrases ayant reçu une probabilité proche entre les différentes classes possibles qui devront faire l'objet d'un étiquetage manuel.

3 Application de l'analyse argumentative au corpus de l'ACL

La méthode développée par Y. Guo et ses collègues semble particulièrement bien adaptée à notre problème. Nous souhaitons en effet catégoriser les termes repérés à l'étape précédente afin notamment d'identifier les méthodes mentionnées dans le corpus ACL Anthology et pouvoir ainsi analyser, par exemple, leur évolution dans le temps. Les termes apparaissant dans des phrases se rapportant au protocole expérimental employé sont donc susceptibles de particulièrement nous intéresser. Il faut noter à ce propos qu'il n'y a pas de frontière étanche entre thèmes et méthodes de recherche dans la mesure où le traitement automatique des langues s'appuie sur ses propres résultats pour concevoir des systèmes en couches empilées : ainsi, un analyseur sémantique reposera fréquemment sur un analyseur syntaxique employé comme outil (et apparaissant donc dans la section méthodologique de l'article).

L'annotation ne porte que sur les résumés des articles. On fait en effet l'hypothèse que les résumés contiennent assez d'information et sont assez redondants pour observer l'évolution du domaine. A l'inverse, aborder l'étude en utilisant le texte complet des articles entraînerait probablement du bruit et complexifierait inutilement les traitements.

Le jeu d'annotation initialement adopté comporte sept catégories différentes et une catégorie AUTRE pour les phrases ne pouvant pas être catégorisées par les étiquettes définies. Ces étiquettes sont les suivantes :

— OBJECTIF : décrit les objectifs de l'article ;
— METHODE : méthodes employées par l'article ;
— RESULTATS : résultats obtenus ;
— CONCLUSION : conclusion de l'article ;
— ARRIERE-PLAN : contexte scientifique ;
— TRAVAUX LIES : positionnement par rapport à des travaux directement liés à ceux présentés ;
— AUTRES TRAVAUX : positionnement par rapport à d'autres travaux.

Ces catégories sont reprises des travaux précédents, notamment (Mizuta *et al.*, 2006; Guo *et al.*, 2011, 2013). Il nous a semblé important de reprendre un jeu de catégories existantes dans la mesure où ces catégories, avec de légères variations, se sont globalement imposées depuis les premiers travaux de S. Teufel. Certaines catégories sont malgré tout peu présentes

dans les résumés de l'ACL Anthology, et finalement quatre catégories transparaissent principalement : les catégories OBJECTIF, ARRIERE-PLAN, RESULTATS et METHODE. Il est rare de trouver des comparaisons avec d'autres travaux dans les résumés de l'ACL Anthology (alors qu'on en trouve fréquemment dans les résumés en biologie par exemple).

Une centaine de résumés d'article issus de l'ACL Anthology ont ensuite été annotés manuellement avec ces catégories (environ 500 phrases, les résumés de l'ACL Anthology étant souvent très courts dans la mesure où il s'agit en grande majorité de résumés d'articles de conférence). Les articles annotés ont été choisis aléatoirement, en prenant soin toutefois qu'ils couvrent différentes périodes et qu'ils contiennent des termes variés. L'annotation a été faite en suivant le guide d'annotation mis au point par Y. Guo, notamment en ce qui concerne les phrases complexes, se rapportant potentiellement à plus d'une catégorie définie (un jeu de préférences est défini pour résoudre ces cas difficiles).

L'algorithme de (Guo *et al.*, 2011) est ensuite repris et adapté à notre cas de figure. L'analyse se fonde en particulier sur les traits positionnels, lexicaux et syntaxiques comme expliqué dans la section précédente. Aucune information spécifique au domaine n'est ajoutée, ce qui rend le processus simple à modéliser et à reproduire. Pour l'analyse syntaxique, le Stanford Parser est utilisé : `http://nlp.stanford.edu/software/lex-parser.shtml` (De Marneffe *et al.*, 2006). On utilise le SVM linéaire tel qu'implémentée dans Weka pour la classification, avec le paramètre -M pour le calcul des probabilités post-apprentissage. On pourra se reporter à (Guo *et al.*, 2011) pour les détails de l'algorithme d'apprentissage et de marquage des phrase selon le modèle défini. Comme résultat, pour chaque phrase du corpus, l'algorithme associe une étiquette choisie parmi les étiquettes possibles.

4 Résultats et discussion

Pour valider les résultats obtenus, un ensemble de résumés est choisi aléatoirement. Les quatre catégories principales sont bien représentées mais inégalement réparties : 18,05 % des phrases sont catégorisées comme ARRIERE-PLAN, 14,35 % comme OBJECTIF, 14,81 % comme RESULTAT et 52,77 %comme METHODE. On voit bien, à la lecture de ces chiffres, l'importance de la dimension méthodologique dans le domaine.

On observe ensuite, pour chaque catégorie possible, le pourcentage de phrases correspondant effectivement à cette étiquette, ce qui permet de mesurer les performances du système en terme de précision. Les résultats obtenus sont présentés dans le tableau suivant.

TABLE 1 – Résultat de l'analyse argumentative (en précision)

Catégorie	Précision
Objectif	83,87 %
Arrière-plan	81,25 %
Méthode	71,05 %
Résultats	82,05 %

Ces résultats sont conformes à l'état de l'art. On voit que les résultats sont globalement satisfaisants, particulièrement en regard du peu de phrases annotées pour l'entraînement. La richesse des traits pris en compte et la stratégie d'apprentissage actif permettent en outre d'avoir des résultats portables d'un domaine à l'autre sans tâche d'annotation lourde. Les résultats sont légèrement moins bons pour la catégorie METHODE car celle-ci est sans doute plus diversifiée que les autres et donc moins facile à cerner.

L'exemple 1 est un texte annoté suite à l'analyse du système (il s'agit de l'article de (Lee *et al.*, 2002), choisi au hasard parmi ceux qui présentent une bonne diversité dans les catégories utilisées). La catégorisation s'effectue au niveau des phrases, ce qui n'est pas sans poser problème : par exemple, dans ce résumé, le fait qu'une méthode hybride est utilisée est indiqué dans une phrase étiquetée OBJECTIF par le système. Les phrases marquées METHODE contiennent toutefois des mots clés précieux, comme *lexical pattern* ou *tri-gram estimation*, ce qui peut permettre d'inférer le fait qu'il s'agit d'un système hybride. On aperçoit au passage des problèmes de numérisation, qui sont typiques du corpus étudié : l'ACL Anthology comprend des textes convertis automatiquement à partir de fichiers PDF de conférences passées, ce qui entraîne parfois des problèmes de qualité.

Exemple 1 : *Un résumé annoté avec l'analyseur de la structure argumentative. Les catégories ajoutées au texte sont*

indiquées en gras.

> Most of errors in Korean morphological analysis and POS (Part-of-Speech) tagging are caused by unknown morphemes . **ARRIERE-PLAN**
> This paper presents a generalized unknown morpheme handling method with POSTAG(POStech TAGger) which is a statistical/rule based hybrid POS tagging system . **OBJECTIF**
> The generalized unknown morpheme guessing is based on a combination of a morpheme pattern dictionary which encodes general lexical patterns of Korean morphemes with a posteriori syllable tri-gram estimation . **METHODE**
> The syllable tri-grams help to calculate lexical probabilities of the unknown morphemes and are utilized to search the best tagging result . **METHODE**
> In our scheme , we can guess the POS's of unknown morphemes regardless of their numbers and positions in an eojeol , which was not possible before in Korean tagging systems . **RESULTATS**
> In a series of experiments using three different domain corpora , we can achieve 97% tagging accuracy regardless of many unknown morphemes in test corpora . **RESULTATS**

Il est dès lors possible d'extraire des mots clés puis de classer ces mots clés suivant les zones considérées. Pour donner un exemple simple d'analyse reposant sur la catégorisation effectuée ici, on peut essayer de tracer l'évolution de différentes tendances dans le temps. Par exemple, pendant la période considérée, les méthodes utilisées ont beaucoup changé, le principal fait marquant étant peut-être le recours massif à l'apprentissage depuis la fin des années 1990. Cette tendance est marquée par un recours quasi systématique dans les articles actuels à des expérimentations donnant lieu à des résultats chiffrés.

Pour confirmer de façon quantitative cette hypothèse, nous nous intéressons à l'évolution dans le temps de la proportion de phrases étiquetées RESULTAT. Sur la figure 1, nous pouvons ainsi observer que la courbe correspondante croit de façon quasi linéaire du début des années 1980 jusqu'à la fin des années 2000.

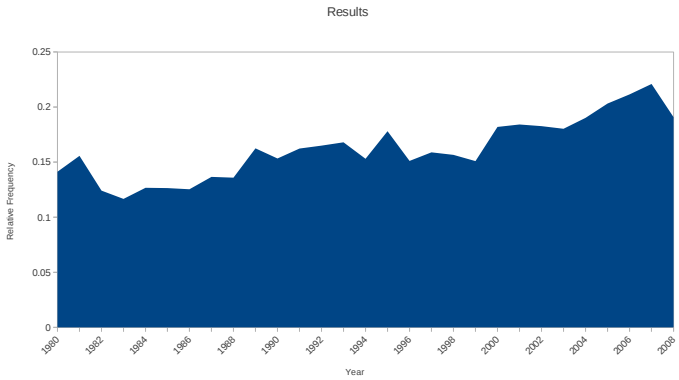


FIGURE 1 – Évolution dans le temps de la proportion de phrases catégorisées par l'outil d'analyse discursive comme étant des phrases concernant des résultats (par rapport au nombre total des phrases contenues dans les articles publiés dans l'année correspondante

Cette illustration relativement simple n'est donnée ici qu'à titre d'exemple. Plus globalement, l'analyse contenue dans cet article ouvre par exemple des perspectives pour une analyse fine de l'évolution des méthodes réellement employées dans les articles, en se focalisant sur la zone étiquetée METHODE (tout en excluant les section ARRIERE-PLAN). Une analyse plus fouillée de l'ACL Anthology en prenant en compte l'analyse argumentative reste à faire, à la suite des travaux de (Anderson *et al.*, 2012) pour les 50 ans de l'ACL.

5 conclusion

Cet article a permis d'appliquer l'analyse argumentative (*text zoning*) à l'ensemble de l'ACL Anthology avec des résultats satisfaisants bien qu'une partie tout à fait minime du corpus initial ait fait l'objet d'une annotation manuelle. On voit ainsi que les techniques d'analyse argumentative, surtout appliquées au domaine bio-médical ces dernières années, sont aussi utilisables dans d'autres cadres et pour d'autres domaines (conformément aux travaux pionniers de S. Teufel) et restent efficaces même pour de grandes masses de documents. Ce type d'analyse devrait permettre d'affiner l'étude des grands corpus scientifiques tant sur le plan historique qu'épistémologique.

Références

ANDERSON A., JURAFSKY D. & MCFARLAND D. A. (2012). Towards a computational history of the acl : 1980-2008. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, p. 13–21, Jeju Island, Corée : Association for Computational Linguistics.

R. E. BANCHS, Ed. (2012). *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Jeju Island, Corée : Association for Computational Linguistics.

DE MARNEFFE M.-C., MACCARTNEY B. & MANNING C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the conference on Language Resource and Evaluation (LREC)*, p. 449–454.

GUO Y., KORHONEN A. & POIBEAU T. (2011). A weakly-supervised approach to argumentative zoning of scientific documents. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 273–283, Edinburgh, Scotland, UK. : Association for Computational Linguistics.

GUO Y., REICHART R. & KORHONEN A. (2013). Improved information structure analysis of scientific documents through discourse and lexical constraints. In *Proceedings of Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, p. 928–937.

LEE G. G., LEE J.-H. & CHA J. (2002). Syllable-pattern-based unknown-morpheme segmentation and estimation for hybrid part-of-speech tagging of korean. *Computational Linguistics*, **28**(1), 53–70.

MIZUTA Y., KORHONEN A., MULLEN T. & COLLIER N. (2006). Zone analysis in biology articles as a basis for information extraction. *International Journal of Medical Informatics*, **75**(6), 468–487.

NOVAK B., MLADENI D. & GROBELNIK M. (2006). Text classi ?cation with active learning. In *From Data and Information Analysis to Knowledge Engineering*, p. 398–405.

RADEV D. R., JING H., STYŚ M. & TAM D. (2004). Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, **40**(6), 919–938.

TBAHRITI I., CHICHESTER C., LISACEK F. & RUCH P. (2006). Using argumentation to retrieve articles with similar citations : An inquiry into improving related articles search in the medline digital library. *I. J. Medical Informatics*, **75**(6), 488–495.

TEUFEL S. (1999). *Argumentative Zoning : Information Extraction from Scientific Articles*. University of Edinburgh.

TEUFEL S. & MOENS M. (2002). Summarizing scientific articles – experiments with relevance and rhetorical status. *Computational Linguistics*, **4**(28), 409–445.

TEUFEL S., SIDDHARTHAN A. & TIDHAR D. (2006). Automatic classification of citation function. In *Proceedings of Empirical Methods in Natural language Processing (EMNLP)*, p. 103–110 : Association for Computational Linguistics.

TONG S. & KOLLER D. (2001). Active Learning with Applications to Text Classi ?cation. *Journal of Machine Learning Research*, **2**, 45–66.

Intégration relationnelle des exemples lexicographiques dans un réseau lexical

Veronika Lux-Pogodalla
CNRS, ATILF, UMR 7118
Nancy, F-54000, France
Veronika.Lux@atilf.fr

Résumé. Nous présentons un ensemble d'exemples lexicographiques intégré dans le Réseau Lexical du Français et explorons son intérêt potentiel en tant que corpus annoté pour la recherche en désambiguisation sémantique automatique.

Abstract. This paper presents a set of lexicographic examples which is being developped along the French Lexical Network. The possibility of using this set as an annotated corpus for research on automatic Word Sense Disambiguation is examined.

Mots-clés : Réseau Lexical du Français, exemples lexicographiques, corpus annoté sémantiquement.

Keywords: French Lexical Network, lexicographic examples, semantically annotated corpus.

Objectifs

Le Réseau Lexical du Français (RL-fr) (Lux-Pogodalla & Polguère, 2011) est élaboré par une équipe de lexicographes à l'ATILF-CNRS [1], selon l'approche de la Lexicologie Explicative et Combinatoire (LEC) (Mel'čuk *et al.*, 1995; Mel'čuk, 2006).

Le RL-fr est développé en même temps que différentes ressources linguistiques (ex. ensemble structuré de caractéristiques grammaticales, hiérarchie d'étiquettes sémantiques, base de fonctions lexicales) qui sont utilisées dans les descriptions lexicographiques du RL-fr. L'ensemble, RL-fr et ressources, forment un *système lexical* dont l'architecture est présentée dans (Polguère, 2009) et (Polguère, 2014). Dans ce système, le RL-fr est donc un graphe dont les nœuds, correspondants aux unités lexicales, sont liés aux éléments d'autres structures, les ressources linguistiques.

En cela l'architecture du RL-fr présentent des points communs avec celle de la base FrameNet, dans laquelle trois composants (*Dictionnaire*, *Base de Frames*, *Phrases d'exemples annotés* [2]) forment *un tout complètement intégré et très densément connecté dans lequel les éléments de chaque composant peuvent pointer vers des éléments des deux autres composants* [3](Baker *et al.*, 1998). Les données de FrameNet peuvent être consultées sur le site `https://framenet.icsi.berkeley.edu/` en utilisant soit un index de Frames, soit un index des Unités Linguistiques, soit un index des textes annotés. De façon comparable, comme tout le système lexical – le RL-fr proprement dit et les ressources linguistiques – est stocké dans une unique base de données SQL(Gader *et al.*, 2012) et comme les liens sont bidirectionnels, plusieurs perspectives différentes sont possibles dans ce système. Ainsi, les données linguistiques qualifiées de *ressources* lorsque nous nous focalisons sur le RL-fr, peuvent, chacune à leur tour, être vue comme *donnée principale* et placée au premier plan.

Dans cet article, après avoir brièvement décrit le RL-fr dans la section 1, nous nous concentrons, dans la section 2, sur une ressource particulière, la collection d'exemples lexicographiques. Dans la section 3, nous faisons une incursion hors du champ de la lexicographie et opérons un complet changement de perspective, en considérant le réseau lexical comme un *graphe d'annotation* et la collection d'exemples, comme un *corpus annoté*.

L'article se veut une invitation, en particulier à l'adresse des chercheurs en désambiguïsation sémantique automatique, à

1. De juin 2011 à octobre 2014, RL-fr est développé dans le cadre du projet de R&D *RELIEF*, financé par l'agence de Mobilisation Économique de Lorraine et par le Fonds Européen de Développement Régional
2. Texte source : *Lexicon*, *Frame Database*, *Annotated Example Sentences*
3. Texte source : *a highly relational and tightly integrated whole : elements in each may point to elements in the other two*

utiliser ce *corpus* qui, comme tout le RLF, sera bientôt mis à disposition de la communauté. Cette démarche a au moins un précédent couronné de succès, puisque les phrases d'exemples annotés de FrameNet ont été utilisées par différentes équipes pour entraîner des systèmes à base d'apprentissage réalisant de l'annotation automatique de rôles sémantiques (Automatic Semantic Role Labeling) [4].

1 Le Réseau Lexical du Français RL-fr

Le RL-fr s'appuie sur les recherches menées antérieurement à Montréal à l'Observatoire de linguistique Sens-texte (OLST) par Alain Polguère et Igor Mel'čuk et en particulier sur l'expérience acquise dans le développement :
– de dictionnaires : Dictionnaire explicatif et combinatoire du français contemporain (Mel'čuk et al., 1984 1988 1992 1999), Lexique actif du français (Mel'čuk & Polguère, 2007), et
– de bases de données lexicales : le DiCo (Steinlin *et al.*, 2005) et ses versions en ligne Dicouèbe (`http://olst.ling.umontreal.ca/dicouebe/`) et Dicopop (`http://olst.ling.umontreal.ca/dicopop/`).
Étant un réseau lexical, le RL-fr est membre de la récente famille des ressources lexicales en *-Net*, comme WordNet (Fellbaum, 1998) ou FrameNet (Baker *et al.*, 2003). Mais la nature des nœuds du RL-fr et surtout la variété de ses liens en font une ressource originale, potentiellement une ressource générique à partir de laquelle générer automatiquement différents dictionnaires : c'est pourquoi le RL-fr appartient à la nouvelle génération des « dictionnaires virtuels » (Selva *et al.*, 2003).

Le RL-fr se présente comme un graphe :
– dont les nœuds sont, essentiellement des unités lexicales [5] , soit lexèmes (ex. VIOLON$_{I}$) soit locutions (⌜accorder ses violons⌝, ⌜pisser dans un violon⌝), mais aussi des clichés linguistiques (*Arrête ton numéro !*) ;
– dont les liens sont, essentiellement, des liens de fonctions lexicales (Mel'čuk, 1996), mais aussi des liens entre copolysèmes et des liens d'inclusion formelle (liens entre une locution et ses composants).
Les fonctions lexicales permettent de modéliser des liens paradigmatiques (relations sémantiques entre unités lexicales) ou syntagmatiques (relations de cooccurrence entre unités lexicales). La figure 1 montre quelques liens de fonctions lexicales entrants et sortants pour le nœud FLÛTE$_{I}$, par exemple, un lien Real$_1$ [6] entre FLÛTE$_{I}$ et JOUER$_{IV.1}$ ainsi qu'entre FLÛTE$_{I}$ et SOUFFLER$_{II}$. La figure 2 montre les copolysèmes (c.-à-d. les autres acceptions, dans un vocable polysémique) du nœud FLÛTE$_{I}$.

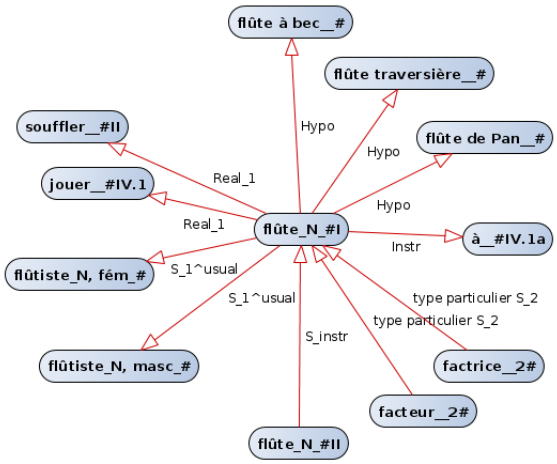


FIGURE 1 – Fonctions Lexicales pour le nœud FLÛTE$_{I}$

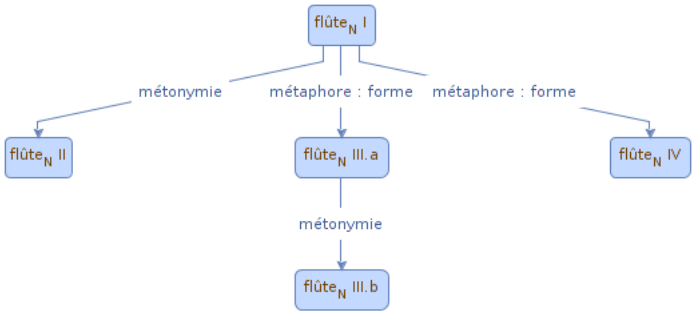


FIGURE 2 – Copolysèmes de FLÛTE$_{I}$

Le RL-fr a commencé avec une nomenclature du français fondamental de 3 800 vocables et son expansion est guidée par certaines fonctions lexicales (Polguère & Sikora, 2013). À terme, notre objectif est de couvrir l'ensemble du français courant. Après presque trois ans de développement, le RL-fr compte environ 22 000 lexies (15 000 vocables) et 42 000 liens.

4. Par exemple, le système Shalmaneser développé à Saarbrücken (`http://www.coli.uni-saarland.de/projects/salsa/shal`), le système LTH, développé à Lund (`http://nlp.cs.lth.se/software` et le système SEMAFOR développé à Carnegie Mellon University (`http://www.ark.cs.cmu.edu/SEMAFOR`)
5. Une unité lexicale correspond à une acception.
6. Real$_1$ lie une unité lexicale L à un verbe de réalisation dont le premier actant est le premier actant de L et dont le deuxième actant est L.

Le travail lexicographique sur RL-fr est largement manuel, ce qui l'apparente au traditionnel artisanat des dictionnaires papier, mais il s'en différencie pourtant radicalement. Le lexicographe ne rédige pas d'articles : il tisse des liens à l'aide d'un éditeur de graphe (Gader *et al.*, 2012). Ce tissage de liens (Polguère, 2012) entre le RL-fr et les ressources linguistiques de description s'analyse aussi comme une annotation, la ressemblance entre système lexical et graphe d'annotation (Bird & Liberman, 2001) ayant étant relevée par (Polguère, 2009). Chaque champ de la description linguistique asssociée à une unité lexicale du RL-fr correspond à une couche d'annotation de l'unité lexicale avec des éléments d'une structure ressource particulière.

Dans le système lexical, le RL-fr est annoté avec des éléments de toutes les structures ressources. À l'inverse, chaque structure ressource se trouve aussi annotée avec des éléments du RL-fr. Ainsi, dans la base de fonctions lexicales, cette annotation associe des exemples aux fonctions lexicales.

Dans la section suivante, nous présentons une ressource particulière, la collection d'exemples lexicographiques.

2 Collection d'exemples lexicographiques

Selon une pratique aujourd'hui standard dans les projets lexicographiques (Atkins & Rundell, 2008; Kilgarriff *et al.*, 2006), les lexicographes du RL-fr illustrent les unités lexicales qu'ils décrivent avec de nombreux exemples, et, en particulier, avec des exemples *authentiques* [7] . Pour le RL-fr, ces exemples sont cherchés dans un corpus dit *corpus-réservoir* [8], qui comprend actuellement : des textes de Frantext (`http://www.frantext.fr`) écrits après 1950, des articles de *L'Est Républicain* (presse quotidienne régionale) écrits entre 1999 et 2002 et FrWac (Baroni *et al.*, 2009), une "tranche" du Web de 2009 [9] . Les lexicographes du RL-fr illustrent chaque unité lexicale avec au moins neuf exemples du corpus-réservoir [10]. Chaque exemple lexicographique est stocké dans la base de données RL-fr avec diverses métadonnées (source, adaptation éventuelle, etc.).

L'ensemble des exemples illustrant des unités lexicales du RL-fr comprend actuellement environ 1 000 000 de mots et 23 000 exemples [11]. Il est une collection d'extraits soigneusement sélectionnés (Atkins & Rundell, 2008; Selva *et al.*, 2009), dans des textes écrits en français (langue source). Ainsi, les exemples lexicographiques sont courts, généralement composés d'une seule phrase. Un "bon" exemple lexicographique contient une occurrence non ambiguë de l'unité lexicale à illustrer. Autant que possible, l'exemple utilise du vocabulaire "simple" – en particulier, un exemple contenant du vocabulaire rare ou spécialisé est inadéquat pour illustrer une unité lexicale courante. De bons exemples pour l'unité lexicale L contiennent, autant que possible la réalisation des actants sémantiques de L ou bien des cibles de Fonctions Lexicales ; l'ensemble des exemples illustre, idéalement, différentes constructions syntaxiques possibles pour L, même si, dans RL-fr, nous ne cherchons pas à illustrer systématiquement *toute la combinatoire d'une unité lexicale* [12], comme le font les auteurs de FrameNet [13].

La collection d'exemples du RL-fr est-elle, de par sa composition, très différente des corpus constitués pour la désambiguïsation sémantique automatique ? Elle ne ressemble certes pas au corpus décrit dans (Rakho *et al.*, 2012), qui se compose de phrases, en français langue source ou traduites en français, extraites de compte-rendus de débats parlementaires et contenant des occurrences de 20 verbes polysémiques particuliers du français. Elle ne ressemble par non plus au corpus de néerlandais décrit dans (Vossen *et al.*, 2011), qui est, essentiellement, une juxtaposition de corpus précédemment constitués (dont le corpus SoNaR comptant 500 millions de mots) et comprend des textes entiers plutôt que des phrases isolées. Cependant, ces deux corpus développés pour la désambiguisation sémantique automatique ne se ressemblent pas entre eux non plus.

7. Soulignons bien que notre démarche n'est pas philologique : nous n'avons pas pour objectif de décrire tous les sens observés en corpus, contrairement à ce qui était le cas dans le *Trésor de la Langue Française*(Dendien & Pierrel, 2003). Aussi notre corpus n'est-il pas à RL-fr exactement ce que Frantext fut au *Trésor de la Langue Française*. Dans notre approche, l'identification des différents sens d'un vocable n'est même pas "dirigée par le corpus", comme elle peut l'être avec un logiciel de traitement de corpus tel que Sketch Engine (Kilgarriff & Kosem, 2012) ; la description de la polysémie repose en premier lieu sur l'intuition du lexicographe. Cependant, si aucun exemple n'est trouvé pour illustrer un sens donné, l'existence de ce sens est évidemment mise en doute.

8. Faute de place, nous n'aborderons pas ici les questions relatives au format des corpus et aux outils de recherche dans ces corpus.

9. FrWac est interrogeable sur : `http://nl.ijs.si/noske/wacs.cgi/first_form`

10. Ils utilisent aussi des exemples extraits de lectures personnelles (livres, journeaux, courriers, Web, etc.) ou entendus à la radio, à la télévision ou dans des conversations, etc. Si nécessaire, les lexicographes adaptent des exemples authentiques, par exemple, en raccourcissant une phrase très longue, en remplaçant un pronom par un groupe nominal, etc.

11. À titre de comparaison : FrameNet compte plus de 170 000 exemples annotés.

12. Texte source : all combinatorial possibilites of the lexical unit

13. Selon : `https://framenet2.icsi.berkeley.edu/docs/r1.5/book.pdf`

3 Changement de perspective : le graphe lexical RL-fr comme ressource d'annotation, la collection d'exemples lexicographiques comme corpus annoté

Chaque texte ajouté à la collection d'exemples lexicographiques RL-fr est annoté avec un lien entre un mot-forme du texte et une unité lexicale du RL-fr. Techniquement, cette collection d'exemples est donc un corpus annoté avec un réseau lexical, et notre architecture n'est pas très différente de celle décrite dans (Vossen *et al.*, 2011).

Le RL-fr est formellement un graphe, ce qui l'apparente à d'autres ressources d'annotation utilisées en désambiguïsation sémantique automatique. En revanche, ressource complètement originale et manuellement développée par une équipe de lexicographes, ainsi qu'expliqué dans la section 1, il se distingue de ressources construites pour la déambiguïsation, plus ou moins automatiquement, en ré-utilisant et en enrichissant des ressources existantes (en particulier, des réseaux de type WordNet comme le font (Vossen *et al.*, 2011) ou (Tsatsaronis *et al.*, 2008)). L'annotation est riche de toute la richesse du RL-fr. Pour un mot-forme annoté du corpus RL-fr, le RL-fr fournit l'ensemble des copolysèmes entre lesquels désambiguiser ainsi que des informations linguistiques formalisées pour chaque copolysème (ex. caractéristiques grammaticales, étiquette sémantique, synonymes, contraires et de nombreux autres liens sémantiques ou de cooccurrence, encodés avec le système des fonctions lexicales, etc.).

Du point de vue de la désambiguisation sémantique automatique, le corpus annoté RL-fr ressemble à un corpus d'évaluation (ou d'entraînement), puisque certains mots-formes du corpus RL-fr, relatifs à des vocables polysémiques donc ambigus dans une analyse automatique, sont *désambiguisés* par une annotation avec une acception particulière [14] . Le corpus RL-fr se présente donc comme une ressource potentielle pour la *désambiguisation sémantique automatique ciblée – targeted WSD*. Par contre, comme l'annotation du corpus RL-fr est relativement peu dense actuellement, le corpus n'est pas adapté pour la désambiguisation sémantique globale – *all-words WSD*, (Navigli, 2009, p. 4).

Faisant l'hypothèse que plus l'annotation est dense plus le corpus annoté RL-fr est une ressource intéressante, nous encourageons la réutilisation des exemples dans RL-fr : un texte du corpus RL-fr peut être réutilisé et illustrer plusieurs unités lexicales, ce qui se traduit par l'annotation de plusieurs mots-formes de ce texte et donc, par une densification de l'annotation globale du corpus RL-fr. Pour faciliter ce processus, le corpus RL-fr est exporté chaque jour de la base de données RL-fr et importé dans TXM (Heiden, 2010). Les lexicographes disposent alors de toutes les fonctionalités de la plateforme TXM pour leurs recherches dans le corpus-RL-fr.

Cependant, chaque texte du corpus est réutilisable en tant qu'exemple lexicographique *seulement jusqu'à un certain point* : un bon exemple E pour l'unité lexicale L_1 ne sera pas réutilisé pour illustrer l'unité lexicale L_2 s'il contient une occurrence ambiguë de L_2 (par exemple, par manque de contexte) ou une occurrence de L_2 sous des traits morpho-syntaxiques rares ou très particuliers, etc. Ainsi, la citation suivante :
À onze heures, avec son ami, il déjeune à la gargote d'une soupe, d'un plat de pommes de terre avec de la viande et de la salade. C'est un festin ! (**Frantext**, FERAOUN Mouloud, *Le fils du pauvre*, 1950, p. 134)
illustre bien DÉJEUNER$_V$ —le régime syntaxique *déjeuner de* est illustré, le second actant de l'unité lexicale est illustré avec plusieurs exemples de plats, etc. Mais le contexte est insuffisant pour savoir si *ami* signifie AMI$_N$ **I.1** ('camarade') or AMI$_N$ **I.2** ('petit ami') c'est pourquoi ce mot-forme ne peut pas être annoté.

Afin d'assurer une bonne qualité lexicographique du RL-fr, seules des occurrences *exemplaires* sont annotées dans le corpus, ce qui limite, *in fine*, la densité d'annotation du corpus-RL-fr. Ici, nos objectifs de lexicographie et les besoins de la désambiguisation sémantique globale ne sont pas convergents.

Si des fins différentes des nôtres nécessitaient d'annoter plus densément le corpus RL-fr, il serait possible, dans notre architecture actuelle, d'associer à une unité lexicale L, un mot-forme dans un texte qui n'aurait pas le statut d'exemple lexicographique pour L – l'annotateur pourrait même préciser dans un commentaire sur l'association pourquoi le texte n'a pas valeur d'exemple pour L. Cependant, l'ambiguité sémantique lexicale étant un phénomène naturel dans la langue, si tous les mots-formes du corpus-RL-fr devaient être annotés, il faudrait au moins prévoir la possibilité d'annoter un mot-forme avec une disjonction de sens et non pas forcément avec un seul sens, comme c'est le cas maintenant.

14. Dans le corpus RL-fr, des mots-formes relatifs à des vocables monosémiques, donc non ambigus, sont aussi annotés, ce qui n'est sans doute pas le cas dans un corpus annoté spécifiquement pour la désambiguisation sémantique automatique.

Conclusions et perspectives

Dans cet article, nous nous sommes concentrés sur la collection d'exemples lexicographiques qui est une des ressources intégrées dans le système lexical RL-fr. Nous avons ainsi illustré les possibilités de changements de perspective – remarquablement faciles – dans le système lexical du RL-fr. Nous avons montré que cette collection d'exemple s'apparente à un corpus annoté potentiellement intéressant hors du projet RL-fr pour des recherches sur la désambiguisation sémantique automatique. Nous avons analysé aussi la mesure dans laquelle nos intérêts dans RL-fr et ceux de cette communauté de recherche sont convergents pour l'annotation du corpus.

L'expérience de l'utilisation du corpus annoté pour entraîner ou évaluer un système de désambiguisation reste encore à réaliser. Des questions pratiques se poseront que nous avons ignorées jusqu'ici : à quel type de système de désambiguisation le corpus annoté convient-il ? le nombre d'entités annotées est-il suffisant pour entraîner un système statistique ? etc.

Et l'utilisation du réseau lexical RL-fr pour la désambiguisation sémantique automatique se présente aussi comme une piste de recherche, d'autant plus intéressante que la désambiguïsation est une difficulté typique de l'analyse automatique, alors que la Lexicographie Explicative et Combinatoire se présente comme orientée vers la synthèse.

Remerciements

Merci à E. Jacquey, S. Ollinger, S. Pescarini, A. Polguère, D. Sikora et trois évaluateurs anonymes pour leurs commentaires constructifs sur une version précédente de cet article. Toutes les erreurs et insuffisances restent bien entendu de ma responsabilité.
Le projet RL-fr/RELIEF est financièrement soutenu par l'Agence de Mobilisation Économique de Lorraine (AMEL) et le Fonds Européen de Développement Régional (FEDER).

Références

ATKINS S. & RUNDELL M. (2008). *The Oxford Guide to Practical Lexicography*. Oxford : Oxford University Press.

BAKER C., FILLMORE C. J. & LOWE J. (1998). The Berkeley Framenet Project. In *Proceedings of the COLING-ACL'98 Conference*, p. 86–90, Montreal, Canada.

BAKER C. F., FILLMORE C. J. & CRONIN B. (2003). The Structure of the FrameNet Database. *International Journal of Lexicography*, **16**(3), 281–296.

BARONI M., BERNARDINI S., FERRARESI A. & ZANCHETTA E. (2009). The WaCky Wide Web : A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, **43**(3), 209–226.

BIRD S. & LIBERMAN M. (2001). A formal framework for linguistic annotation. *Speech Communication*, **33**(1-2), 23–60.

DENDIEN J. & PIERREL J.-M. (2003). Le Trésor de la Langue Française informatisé : un exemple d'informatisation d'un dictionnaire de langue de référence. *Traitement Automatique des Langues (T.a.l.)*, **44**(2), 11–37.

C. FELLBAUM, Ed. (1998). *WordNet : An Electronic Lexical Database*. Cambridge, MA : The MIT Press.

GADER N., LUX-POGODALLA V. & POLGUÈRE A. (2012). Hand-crafting a lexical network with a knowledge based graph editor. In *Proceedings of CogALex-III Workshop of the 24th International Conference on Computational Linguistics (COLING2012)*, p. 109–125, Mumbai.

HEIDEN S. (2010). The TXM platform : Building Open-Source Textual Analysis Software Compatible with the TEI Encoding Scheme. In *Proceedings of the 24th Pacific Asia Conference on Language Information and Computation*, Sendai, Japan : ENS-Lyon.

KILGARRIFF A. & KOSEM I. (2012). Corpus tools for lexicographers. In S. GRANGER & M. PAQUOT, Eds., *Electronic lexicography*, p. 31–155. Oxford : Oxford University Press.

KILGARRIFF A., RUNDELL M. & DHONNCHADHA E. (2006). Efficient corpus development for lexicography : building the New Corpus for Ireland. *Language Resources and Evaluation*, **40**(2), 127–152.

LUX-POGODALLA V. & POLGUÈRE A. (2011). Construction of a French Lexical Network : Methodological Issues. In *Proceedings of WoLeR 2011 International Workshop on Linguistic Resources, ESSLLI (European Summer School in Logic, Language and Information*, Ljubljana.

MEL'ČUK I. (1996). Lexical Functions : A Tool for the Description of Lexical Relations in the Lexicon. In L. WANNER, Ed., *Lexical Functions in Lexicography and Natural Language Processing*, p. 37–102. Amsterdam/Philadelphia : Benjamins.

MEL'ČUK I. (2006). Explanatory Combinatorial Dictionary. In G. SICA, Ed., *Open Problems in Linguistics and Lexicography*, p. 225–355. Monza : Polimetrica.

MEL'ČUK I., CLAS A. & POLGUÈRE A. (1995). *Introduction à la lexicologie explicative et combinatoire*. Paris/Louvain-la-Neuve : Duculot.

MEL'ČUK I. & POLGUÈRE A. (2007). *Lexique actif du français. L'apprentissage du vocabulaire fondé sur 20 000 dérivations sémantiques et collocations du français*. Champs linguistiques. Brussels : De Boeck & Larcier.

MEL'ČUK ET AL. I. (1984, 1988, 1992, 1999). *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques. Volumes I–IV*. Montréal : Les Presses de l'Université de Montréal.

NAVIGLI R. (2009). Word sense disambiguation : a survey. *ACM Computing Surveys*, **41**(2), 1–69.

POLGUÈRE A. (2009). Lexical systems : graph models of natural language lexicons. *Language Resources and Evaluation*, **43**(1), 41–55. Springer.

POLGUÈRE A. (2012). Like a lexicographer weaving her lexical network. In *Proceedings of CogALex-III Workshop of the 24th International Conference on Computational Linguistics (COLING2012)*, p. 1–4, Mumbai.

POLGUÈRE A. (2014). Principes de modélisation systématique des réseaux lexicaux. In *Actes de la 21ème conférence Traitement Automatique des Langues Naturelles*, Marseille.

POLGUÈRE A. & SIKORA D. (2013). Modèle lexicographique de croissance du vocabulaire fondé sur un processus aléatoire, mais systématique. In C. R. C. MASSERON, C. GARCIA-DEBANC, Ed., *Enseigner le lexique. Pratiques sociales, objets à enseigner et pratiques d'enseignement*, collection de l'AiRDF 5. Laval, Québec : Presses de l'Université Laval.

RAKHO M., ÉRIC LAPORTE & CONSTANT M. (2012). A new semantically annotated corpus with syntactic-semantic and cross-lingual senses. In N. C. C. CHAIR), K. CHOUKRI, T. DECLERCK, M. U. DOĞAN, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Eds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey : European Language Resources Association (ELRA).

SELVA T., VERLINDE S. & BINON J. (2003). Vers une deuxième génération de dictionnaires électroniques. *Traitement Automatique des Langues (T.A.L.)*, **44**(2), 177–197.

SELVA T., VERLINDE S. & BINON J. (2009). Les bases de données au service d'un dictionnaire d'(auto-)apprentissage pour allophones. *Lexique*, **19**, 181–214.

STEINLIN J., KAHANE S. & POLGUÈRE A. (2005). Compiling a "classical" explanatory combinatorial lexicographic description into a relational database. In *Proceedings of the Second International Conference on the Meaning Text Theory (MTT'2005)*, p. 477–485, Moscow.

TSATSARONIS G., VARLAMIS I. & VAZIRGIANNIS M. (2008). Word Sense Disambiguation with Semantic Networks. In P. S. ET AL., Ed., *Text, Speech and Dialog - proceedings of the 11th international conference TSD*, volume 5246 of *Lectures Notes in Computer Science*, p. 219–226, Berlin-Heidelberg : Springer-Verlag.

VOSSEN P., GÖRÖG A., LAAN F., VAN GOMPEL M., IZQUIERDO R. & VAN DEN BOSCH A. (2011). Dutchsemcor : Building a semantically annotated corpus for dutch. In *Proceedings of eLex 2011*, p. 286–296.

Les couleurs des gens

Mathieu Lafourcade1, Nathalie Le Brun2, Virginie Zampa3
(1) Lirmm, Université Montpellier 2, France
(2) Imagin@t, 34400 Lunel
(3) Lidilem, Grenoble 3 BP25, 38040 Grenoble cedex 9, France
mathieu.lafourcade@lirmm.fr, imaginat@imaginat.name, virginie.zampa@u-grenoble3.fr

Résumé. En TAL et plus particulièrement en analyse sémantique, les informations sur la couleur peuvent être importantes pour traiter correctement des informations textuelles (sens des mots, désambiguïsation et indexation). Plus généralement, connaître la ou les couleurs habituellement associée(s) à un terme est une information cruciale. Dans cet article, nous montrons comment le *crowdsourcing*, à travers un jeu, peut être une bonne stratégie pour collecter ces données lexico-sémantiques.

Abstract. In Natural Language Processing and semantic analysis in particular, color information may be important in order to properly process textual information (word sense disambiguation, and indexing). More specifically, knowing which colors are generally associated to terms is a crucial information. In this paper, we explore how crowdsourcing through a game with a purpose (GWAP) can be an adequate strategy to collect such lexico-semantic data.

Mots-clés: association couleur-mot, réseau lexical, crowdsourcing.
Keywords: Word Color Associations, Lexical Network, Crowdsourcing

1 Introduction

La couleur, en tant qu'élément de notre vie quotidienne, est une donnée intéressante en Traitement Automatique du Langage Naturel. En effet, fournir des informations relatives aux associations mots-couleurs, en plus des données classiques (hyperonymes, parties de, rôle sémantique, etc.) à un système dédié à l'analyse sémantique de textes, pourrait en améliorer grandement les performances. Bien que ce soit une observation marginale par rapport au sujet de cet article, il existe indiscutablement une très forte liaison entre couleurs et émotions. Ainsi, si on relève souvent des associations entre des couleurs et des termes abstraits relatifs à des émotions (comme la crainte, la colère, le danger, l'espoir, etc.), c'est encore plus vrai pour les termes désignant des choses concrètes (comme le ciel, un lion, la mer, la neige, etc.).

En psychologie, de nombreuses études concernent les associations entre mots et couleurs et leur impact sur la communication (verbale ou non-verbale). Saif (2011a) montre que la notion de couleur est d'une importance capitale pour la qualité du message délivré, que ce soit en marketing (Sable and Akcay, 2010), en conception de sites web (Meier, 1988; Pribadi et al., 1990), ou pour caractériser visuellement une information (Christ, 1975; Card et al.,1999). La couleur est indiscutablement une donnée très porteuse de sens. Mais pour de nombreux chercheurs, les significations attribuées aux couleurs dépendent de plusieurs facteurs. Luscher (1969), psychothérapeute auteur du test éponyme, le *Lüscher color test*, un outil qui permet d'évaluer l'état émotionnel d'une personne en fonction de ses préférences de couleurs, affirme que le choix de telle ou telle couleur par une personne dépend directement de (voire traduit) ses émotions et son état psychique. Child et al. (1968), et Ou et al. (2011) montrent que les préférences en matière de couleurs varient en fonction de l'âge et du sexe. De plus, pour certaines couleurs, on rencontre des variations lexicales en fonction des langues et des cultures (par exemple, en turc et en hongrois, il existe deux mots différents pour *rouge*).

Savoir si la relation établie entre couleur et sens est indépendante de l'âge, du sexe, ou de la nationalité, reste une question très controversée qui fait actuellement l'objet d'un vif débat. Cela pourrait être le cas, par exemple, pour la relation entre la couleur *rouge* et la notion de *danger*, étant donné que le *rouge* est associé aux idées de *sang* et de *feu* dans de nombreuses langues/cultures. Berlin and Kay (1969) disent que les différences pourraient être classées hiérarchiquement, et qu'on retrouve un nombre limité de termes de couleurs de base dans différentes cultures. Cette analyse est issue d'une comparaison des mots désignant des couleurs dans 20 langues, mais elle est particulièrement controversée, en particulier concernant la méthodologie employée pour la collecte des données. Dans un même ordre d'idée, on pourrait remarquer que de nombreuses expressions faisant référence à une couleur ont le même sens dans différentes langues, *a fortiori* si elles sont culturellement proches, ce qui n'est guère surprenant. Par exemple, *dark thoughts* en anglais et *idées noires* en français ont des sens pratiquement équivalents, de même que *to see red* et *voir rouge*. Cependant, il s'agit évidement d'exemples issus de langues tellement proches linguistiquement et culturellement, que cela ne peut guère valider une quelconque universalité des associations possibles entre lexiques et couleurs.

De nombreuses études, principalement en anglais, visent à établir des relations entre couleurs et mots, et en particulier couleurs et émotions. La plupart sont effectuées d'une manière classique en psycholinguistique, et leurs données brutes sont généralement de taille modeste et non librement accessibles. De plus, comme nous l'avons souligné précédemment, il est extrêmement délicat (et probablement imprudent) d'extrapoler directement le résultat de telles études d'une langue à une autre. Ainsi, il n'est actuellement pas possible d'aboutir à un consensus sur l'universalité des associations de mots et de couleurs quand il s'agit de sentiments, ou du moins de termes abstraits. Saif (2011b), au terme de diverses expériences sur un échantillon de 11 couleurs, a conclu que plus de 30% des termes issus d'un dictionnaire de vocabulaire courant étaient fortement liés à une couleur. Environ 33% des catégories de thésaurus (comme Roget) révèlent des associations de couleurs, et les termes abstraits sont associés à des couleurs (principalement de façon métaphorique) presque aussi souvent que ceux désignant des objets, des entités physiques. De même, Nijdam (2010) compare différents modèles issus de la recherche sur les liens entre couleurs et émotions, et propose une correspondance. Il conclut que si certains modèles de signification des couleurs ont des points communs et se recoupent partiellement, ils témoignent aussi d'une forte variabilité, sans doute parce que la couleur est appréhendée différemment suivant la situation affective/culturelle du sujet. En résumé, l'acquisition de données lexicales sur les associations mots-couleurs se heurte à une forte variabilité, même au sein d'une même langue: plus le terme est abstrait, plus on doit s'attendre à une variabilité des couleurs qui lui sont associées.

Ozbal et al. (2011) sont à l' origine d'une ressource qui répertorie des associations mots-couleurs en anglais. Sur une sélection de 200 mots (un sous-ensemble des mots utilisés dans Grefenstette (2005)), ils ont utilisé trois méthodes automatisées pour comparer leur ressource avec les associations faites par le service Amazon Mechanical Turk (10 annotateurs-11 couleurs): analyse d'images, modèles de langages sur les données du web, et similarité entre mots et couleurs en utilisant LSA. Actuellement, pour autant que nous le sachions, Ozbal et al. (2011) sont les seuls à avoir tenté de réaliser une ressource lexicale d'associations mots-couleurs en anglais. Ce type de ressource est quasiment inexistant dans d'autres langues, et notamment, il n'existe rien de similaire en français. Cela dit, comme l'a fait remarquer Grefenstette (2005), *s'il y a une caractéristique qu'en général les gens connaissent à propos des choses (concrètes ou même abstraites), c'est bien leur couleur typique*. C'est pourquoi cette information est généralement absente des dictionnaires et autres ressources lexicales, alors qu'elle serait d'un intérêt majeur pour de nombreuses applications informatiques, et en particulier en TAL.

L'information de couleur peut être très utile dans le cadre de la désambiguïsation lexicale automatique (*Word Sense Disambiguation – WSD*). Par exemple, le mot *tissu* est polysémique et peut signifier –entre autres- *étoffe* ou *tissu vivant*. Si, dans un texte, le mot *tissu* est associé à *bleu*, cette information va pouvoir aider à choisir la signification appropriée, ou du moins permettre d'éliminer celle(s) qui est (sont) non pertinente(s). L'information relative à la couleur est également précieuse dans la situation où on cherche à retrouver un mot momentanément oublié mais qu'on a *sur le bout de la langue*, à l'aide d'une application informatique à laquelle on fournit des indices (Lafourcade 2012 et Joubert 2012).

L'objectif de cet article est de montrer comment on peut générer efficacement et relativement vite une ressource lexicale d'associations entre couleurs et mots : c'est pour acquérir ce type de données que nous avons mis au point un GWAP (Game With A Purpose selon le concept défini par L. von Ahn. (2006)), baptisé ColorIt (accessible à http://www.jeuxdemots.org/colorit.php). Dans un premier temps nous évoquons l'intérêt que peut avoir l'information relative à la couleur dans le cadre de la *WSD*, puis nous exposons rapidement le fonctionnement du jeu ColorIt et poursuivons par une analyse détaillée des résultats obtenus.

2 ColorIt, un jeu pour associer des couleurs et des mots

Le but de ColorIt est de collecter des associations spontanées entre couleurs et termes, qu'il s'agisse de couleurs attribuées à des objets concrets, ou associées symboliquement et subjectivement à des verbes ou noms désignant des entités abstraites.

2.1 Intérêt des associations mots-couleurs pour la désambiguïsation lexicale

L'un des intérêts majeurs d'une ressource lexicale rassemblant des associations entre mots et couleurs est son exploitation dans le contexte de la désambiguïsation lexicale automatique (*WSD*). En effet, dans un texte donné, la précision apportée par l'indication de couleur est souvent le détail qui va permettre de sélectionner le sens approprié d'un terme polysémique. Dans de nombreux cas, l'association à une couleur spécifique est importante. Bien sûr, la technique a ses limites : il existe des mots pour lesquels l'information de couleur est sans objet (*sans couleur*), d'autres auxquels on peut associer de nombreuses couleurs (comme *voiture*), ou qui peuvent être simultanément de plusieurs couleurs (noir et blanc pour un *pingouin*, multicolore pour un *arc-en-ciel*). Sans prétendre à l'exhaustivité, nous donnons dans le tableau ci-après quelques mots pour lesquels l'information de couleur peut aider à lever l'ambiguïté résultant de la polysémie (les sens que l'on peut éliminer sont marqués par *, les ambiguïtés qui subsistent par ?). Les résultats présentés ci-dessous sont ceux obtenus par désambiguïsation endogène d'un terme associé à un contexte (la couleur) sur les données du réseau lexical JeuxDeMots.

lit blanc	lit > couche: blanc, rouge, * lit>rivière	**regard bleu/azur** **regard noir**	regard> expression (symbolique) * regard>trou
trompe grise **trompe dorée**	trompe>éléphant *? trompe>instrument de musique * trompe>anatomie humaine	**robe rose / bleu** **robe blanche /jaune** **robe pie/alezane**	robe>vêtement * robe>pelage (animal) ? ambigu pour certaines couleurs
bas noir	bas> sous-vêtement * bas>partie inférieure	**tableau noir**	tableau>école ? tableau>œuvre d'art (improbable)
manteau rouge	manteau>vêtement * manteau>géologie	**feuille blanche** **feuille rouge/jaune/vert**	feuille>papier (le plus probable) *? feuille>arbre (improbable) ? ambigu pour certaines couleurs
ours blanc/brun **ours rose**	ours>animal / peluche ours>peluche * ours>imprimerie	**canapé bleu**	canapé>meuble * canapé>petit four ? ambigu pour certaines couleurs
piste blanche/ verte / rouge	piste>ski (symbolique) *piste>chemin	**brioche dorée**	brioche>viennoiserie *? brioche>ventre
langue blanche / rose	langue>organe * langue>langage	**savon noir / rose / blanc**	savon>savonnette *savon>réprimande
sucre blanc/roux **sucre gris**	sucre>substance *? sucre>électricité (ambigu mais improbable)	**culotte bleu / rouge/ blanche**	culotte>sous vêtement * culotte>boucherie * culotte>défaite
costume anthracite	costume>tenue (le plus probable) *? costume>déguisement	**motif fleuri**	motif>dessin *motif> raison

Table 1: Dans environ 70 % des cas, l'information de couleur permet de déduire le sens convenable et d'éliminer les autres sans ambiguïté. Parfois, une incertitude demeure, mais un niveau de confiance peut être calculé.

2.2 Principe du jeu

Un terme est proposé au joueur avec un choix réduit de couleurs *via* une palette. Il doit cliquer sur la couleur qu'il associe au mot. Il peut également cliquer sur *sans couleur* ou "passer" (s'il ne connait pas le mot par exemple). Passer n'est pas pénalisant. Enfin, si aucune des couleurs de la palette ne convient, ou s'il souhaite en mettre plus d'une, il peut la ou les saisir dans le champ de texte. Le terme présenté peut être ambigu (*feuille)* ou contextualisé (*feuille>papier* / *feuille>plante*), le réseau JDM contenant pour chaque terme polysémique ses raffinements.

Une fois que le joueur a validé son choix, une notification apparait avec son score ainsi que les réponses données par les autres joueurs pour ce mot. Le score dépend de l'adéquation entre la réponse du joueur et la distribution des couleurs déjà affectées au mot *via* les réponses des autres. Il existe deux types de réponses : **une ou plusieurs couleurs** ou bien **sans couleur**. Si la réponse du joueur correspond à une des plus fréquentes, il gagne des points, sinon il en perd. Le nombre de points gagnés est covariant avec le poids des couleurs liées au mot dans le réseau lexical selon une fonction logistique sigmoïde *S*. Si une couleur est faiblement associée (par rapport à d'autres), le joueur va gagner peu de points. Par contre si elle est très fortement associée, le gain sera plus grand mais plafonné. Le poids associé à la relation mot-couleur est incrémenté de 1 à chaque réponse. Une relation de couleur nouvellement associée rentre dans le réseau lexical avec un poids de 1. Supposons par exemple, que dans le réseau le terme *éléphant* soit associé à *gris* avec un poids de p1 et à *blanc* avec un poids de p2. Si le joueur sélectionne *gris* il gagnera *S(p1)* points et le poids de la relation *gris* passera à p1+1. Si le joueur clique sur *pas de couleur* il perd *S(p1+p2)* points mais la relation *pas de couleur* est introduite et prend la valeur 1. Il n'y a pas d'avantage particulier à proposer des couleurs plutôt que *pas de couleur*.

Ce système incite les joueurs à être honnêtes en sélectionnant des couleurs appropriées, et à essayer d'être originaux quand c'est possible. L'honnêteté présupposée de la grande majorité des joueurs a été vérifiée empiriquement et se révèle être une constante dans les GWAP pour lesquels nous avons ce type d'information (Chamberlain, 2013). D'une part une erreur peut être coûteuse en termes de points, d'autre part un joueur qui chercherait à saboter les données en répondant systématiquement de manière incorrecte se lasserait très vite. Aucune tentative de sabotage n'a été détectée à ce jour. Tous les 100 points, le joueur passe au niveau supérieur et bénéficie d'un plus grand nombre de caractères autorisés dans le champ de texte libre. Les facteurs qui rendent le jeu motivant sont la possibilité de comparer sa (ses) réponse(s) avec celles des autres joueurs et la progression en termes de niveau, puisque plus le niveau est élevé plus on peut saisir de mots dans le champ de texte libre. L'expérience montre que la plupart des joueurs préfèrent entrer plusieurs réponses de couleur dans le champ de texte que se contenter de cliquer sur une des couleurs proposées. ColorIt s'avère être un jeu stimulant et très original si l'on en croit les nombreux retours positifs des joueurs. La variété du vocabulaire proposé et la présence de termes ambigus ou contextualisés (*ours>animal, frégate>oiseau, magistrat>singe, avocat>fruit, avocat>justice)* évite la monotonie. A côté de certains mots dont la couleur est évidente (comme *neige*, *nuit*, *charbon*, etc.) et ne se prête pas ou peu à une interprétation, d'autres peuvent avoir une large gamme de couleurs possibles, qu'elles soient objectives pour des noms concrets (comme *fleur*, *voiture*, etc.) ou subjectives pour des concepts abstraits (comme *colère*, *tristesse*, etc.), ou pas de couleur du tout (comme *augmentation*, *audace*, etc.). Choisir la couleur que l'on pense être la plus pertinente n'est pas une tâche facile et la confrontation avec les réponses des autres génère excitation, suspense, et… bonne ou mauvaise surprise ! Bien que le jeu ait pour but la

collecte de données lexico-sémantiques, aucune connaissance linguistique n'est requise pour jouer, et les données recueillies sont de bonne qualité (cf section 3) et parfois d'un niveau de précision et de nuance remarquable (voir liste des couleurs/apparences sur le site à jeuxdemots.org/colorit.php?action=colorlist). Ce niveau de précision va sans doute bien au-delà de ce qui est nécessaire pour de la désambiguïsation lexicale, mais reste parfaitement justifié en représentation des connaissances.

2.3 Dans les coulisses

Le projet JDM (Lafourcade, 2007 and Chamberlain, 2013) a pour but de construire un réseau lexical et sémantique le plus large possible par le biais de jeux (GWAP) et de *crowdsourcing*. Un tel réseau est composé de termes (les nœuds ou sommets) et de relations entre ces nœuds. Les relations sont typées, orientés et pondérées. La ressource contient actuellement plus de 320 000 termes et 9 millions de relations. Le réseau lexical JDM contient des termes, des significations de termes (raffinements), et différentes informations sémantiques (comme *personne*, *être vivant*, *abstrait*, *concret*, *artefact* etc.). Les relations sont pondérées (les plus évidentes ont les poids les plus élevés). Les relations affectées d'un poids négatif soulignent une impossibilité intéressante, comme une exception (par exemple: *autruche* agent: -100 *voler*). On notera qu'une telle approche peut être utilisée pour désigner des impossibilités en termes de couleurs. Les différentes significations d'un terme sont liées au terme principal par des *raffinements*. Par exemple, *frégate* est liée à *frégate>navire* et *frégate>oiseau*. Le raffinement *frégate>navire* est lui-même raffiné en *frégate>navire>moderne* et *frégate>navire>ancien*. Chaque raffinement ou sous-raffinement est lui-même connecté à d'autres termes dans le réseau.

Dans ce réseau, l'information relative à la couleur n'était jusqu'à maintenant présente qu'à travers la relation *caractéristique*, donc mélangée à des attributs de nature variée. Etant donnée l'importance de cette relation en représentation des connaissances, elle fait désormais l'objet d'une relation spécifique que le jeu ColorIt permet d'enrichir de manière conséquente. Un algorithme heuristique a été conçu de manière à avoir une probabilité raisonnable de sélectionner un terme pour lequel l'information de couleur a du sens. Ainsi, nous évitons de lasser les joueurs en leur proposant trop de termes pour lesquels c'est la réponse *pas de couleur* qui s'impose. Cet algorithme qui choisit le terme à proposer au joueur est le suivant : nous sélectionnons aléatoirement, dans le réseau JeuxDeMots, un mot qui a au moins une relation de couleur positive. Nous proposons ensuite au joueur, *via* un *pile ou face* virtuel, soit ce terme soit un de ses voisins dans le réseau. Ceci entraine une propagation rapide de l'acquisition des relations de couleur à travers le réseau. Un terme qui aura été caractérisé comme *sans couleur* plusieurs fois sera retiré de la liste des voisins sélectionnables. De la même manière, les raffinements des termes "colorés" deviennent rapidement "colorés" eux-mêmes. Le principe qui sous-tend cet algorithme de propagation est qu'il y a plus de chances de sélectionner un terme éligible pour l'information de couleur s'il est lié à un mot qui a déjà lui-même une caractéristique de couleur. Une sélection entièrement aléatoire serait contre-productive parce que la plupart du temps, aucune information de couleur ne pourrait être proposée par le joueur; en effet nous avons estimé par échantillonnage qu'environ 10% seulement des quelques 320 000 termes du réseau sont éligibles pour l'information de couleur.

Si une réponse fournie par le joueur existe dans le réseau mais n'est pas caractérisée comme une couleur ou une apparence, elle est alors proposée et sera validée (ou invalidée) en tant que telle par un administrateur. Si par contre le terme proposé n'existe pas dans le réseau, il doit d'abord y rentrer (toujours par validation), avant d'être caractérisé comme une couleur ou une apparence. L'interface Diko du réseau lexical JDM est un outil contributif permettant la validation des termes proposés (http://www.jeuxdemots.org/diko.php). L'algorithme de propagation a été amorcé avec les termes de couleurs proposés dans le jeu : *blanc, gris, noir, rouge, rouge foncé, orange, jaune, vert clair, vert, bleu, bleu clair, violet, rose, beige, marron.* Avant la création de ColorIt, ces couleurs étaient associées à de nombreux termes *via* d'autres relations que la couleur.

3 Collecte et évaluation des données

En neuf mois (août 2013 à avril 2014), plus de 28 000 associations mots-couleurs ont été créées et plus de 16 000 termes ont été dotés d'une information de couleur. Ainsi, en moyenne, un terme est associé à 2,2 couleurs (hors *sans couleur*). Le nombre total de votes de couleurs a excédé 192 000, le nombre de joueurs n'est pas connu avec précision (il n'est pas nécessaire de s'enregistrer pour jouer). D'après notre estimation sur le nombre de termes éligibles pour une information de couleur, nous pensons qu'actuellement la moitié des termes éligibles ont été dotés d'une information de couleur (soit environ 16 000 termes sur 10 % de 320 000).

Dès qu'une couleur a été associée à un terme, le réseau est mis à jour avec cette couleur ou apparence. Par exemple, le mot *eau* est associé avec différents bleus (*bleu clair* (26), *bleu* (9), *bleu lagon* (1), *turquoise* (1), etc.) plusieurs autres couleurs (*verdâtre* (4), *vert* (2), etc.) et différentes apparences (*trouble* (20), *transparent* (9+9), *limpide* (4), *incolore* (4), etc.). La possibilité d'ajouter des couleurs ou des apparences *via* le champ de texte libre permet de compléter le réseau avec des noms de couleurs spécifiques comme *jaune chartreuse, mandarine*, etc., des apparences comme *rayé*, *translucide*, etc., ou encore de nouvelles combinaisons de couleurs (la liste des couleurs est disponible à l'url http://www.jeuxdemots.org/colorit.php?action=colorlist). Actuellement, plus de 700 termes correspondent à des couleurs, des associations de couleurs ou des apparences.

Les données collectées ont été évaluées manuellement, sur deux échantillons d'environ 500 termes, représentant à peu près 2 500 associations mots-couleurs. Le premier échantillon S_C a été prélevé parmi les termes ayant au moins une association de couleur, le second S_{NC} parmi ceux ayant au moins un vote de type *sans couleur*. Les termes de l'échantillon S_C peuvent avoir des associations de type *sans couleur*, et vice-versa. Nous avons demandé à des volontaires de contrôler au hasard certaines associations mots-couleurs, et de dire si elles étaient correctes ou non, que ce soit dans le cas d'association de type *couleur* ou *sans couleur*. Cette méthode d'évaluation, où l'on demande à des volontaires de juger des associations (tâche fermée) est orthogonale avec la méthode d'acquisition où l'on demande aux gens de produire les associations (tâche ouverte). Notons que la méthode d'évaluation ne permet pas de se prononcer sur les associations de couleurs éventuellement omises (les silences).

	S_C Associations de couleur correctes	**S_C Associations de couleur incorrectes**	**S_{NC} Associations sans couleur correctes**	**S_{NC} Associations sans couleur incorrectes**	**Total**
# votes	13 308	366	708	81	14 463
%	92 %	2,5 %	5 %	0,5 %	100%
	sous-total 13308/13674 = 97%		sous-total 708/789 = 90%		

Table 2. Evaluation des associations mots-couleurs et mots-*sans couleur*.

L'évaluation par quelques dizaines de volontaires (*via* le jeu Askit : http://www.jeuxdemots.org/askit.php) a couvert environ 10 % des associations mots-couleurs générées par le jeu (acquisition). Les données collectées semblent contenir un pourcentage assez bas d'associations incorrectes (cf. Table 2). En outre, un examen plus approfondi du réseau lexical montre que les associations incorrectes sont affectées d'un poids faible par rapport aux associations correctes. Par exemple, pour le terme *soleil*, on trouve une fois l'association avec *bleu* pour 100 fois l'association avec *jaune*. De plus, l'examen des termes polysémiques montre que les couleurs associées aux raffinements sont le plus souvent adéquates et pourront éventuellement être discriminantes dans le cadre de la désambiguïsation lexicale (*WSD*).

Termes liés à	**politique (2702 termes)**	**zoologie (3191 termes)**	**arts (2815 termes)**	**émotions (304 termes)**	**botanique (4481 termes)**
Accord entre joueurs	0.55	0.65	0.71	0.28	0.76

Table 3. Kappa moyen entre les joueurs en fonction de différents champs sémantiques d'intérêt concernant les couleurs (réelles ou symboliques). Le chiffre entre parenthèses indique le nombre de termes reliés à ce champ sémantique dans le réseau lexical. Manifestement, les couleurs sont difficiles à accorder aux émotions.

La Table 3 présente l'évaluation de l'accord entre joueurs en fonction de l'appartenance des termes à tel ou tel domaine (champ sémantique). La valeur de l'accord pour un terme donné est la moyenne du Kappa de Cohen entre les associations des joueurs prises 2 à 2 pour ce terme. La valeur de l'accord pour un champ sémantique est la moyenne des accords des termes de ce champ qui ont été joués. Certains termes peuvent appartenir à plus d'un champ. C'est pour le champ sémantique relatif aux plantes que l'accord entre joueurs est le plus élevé. Il semble que certains joueurs se réfèrent à des sources extérieures (comme par exemple Wikipédia) pour savoir quelle(s) couleur(s) affecter à une plante ou un animal inconnu, jouant ainsi le rôle d'identificateurs d'informations qu'il serait très difficile d'extraire automatiquement (communication personnelle avec certains des joueurs). L'accord le plus faible est celui relatif au champ sémantique des émotions, ce qui s'explique aisément par le haut degré de subjectivité des associations correspondantes (et ceci bien que notre expérience n'ait été réalisée qu'en français, et sur une population relativement homogène constituée de 70 % de femmes entre 30 et 50 ans). Concernant le champ sémantique des arts, on trouve beaucoup de noms de couleurs exprimés au moyen d'un vocabulaire recherché, mais qui une fois décryptés présentent finalement assez peu de variations.

4 Conclusion

Nous avons conçu un jeu simple mais stimulant et efficace pour collecter des données relatives aux associations termes-couleurs. En quelques mois plus de 15 000 termes ont été associés à une ou des couleurs et environ 4 000 ont été caractérisés comme *sans couleur*. De nombreuses couleurs et apparences ont été introduites dans le réseau lexical JeuxDeMots *via* le jeu ColorIt. A notre connaissance, les données collectées constituent la première ressource (dynamique et libre) en français pour ce type d'association. L'une des difficultés dans la conception du jeu a été l'algorithme de propagation permettant de sélectionner des termes éligibles à une information de couleur, de façon à ne pas lasser les joueurs en leur proposant trop de termes auxquels il n'est pas pertinent d'attribuer une couleur.

Nous avons eu plusieurs surprises avec les données collectées grâce à ColorIt. Premièrement, le choix *sans couleur* a été initialement conçu pour servir d'échappatoire aux joueurs lorsque rien ne convenait. Mais il s'est rapidement avéré que la caractéristique *sans couleur* était précieuse pour éliminer certaines significations en cas de polysémie. En effet, dans le cadre de la désambiguïsation lexicale, la sélection de sens par les indications de couleur est plus souvent réalisée par élimination que par sélection ; cela s'explique, du moins partiellement, par la polysémie par métaphore: des sens abstraits (donc *sans couleur*) dérivent souvent de termes concrets. Deuxièmement, un effet collatéral et néanmoins positif du jeu a été d'enrichir le réseau JDM d'un grand nombre de couleurs et apparences qui n'y figuraient pas encore. En plus de vraies couleurs nouvellement introduites, le réseau contient maintenant de nombreux termes relatifs à l'aspect visuel, comme *translucide*, *trouble*, *rayé*, *tacheté*, etc.

Pour aller plus loin, une des perspectives est d'évaluer la corrélation entre couleur et sentiment associés à certains termes, en particulier les noms abstraits. Dès que nous disposerons d'un nombre suffisant d'associations mots-couleurs, nous pourrons envisager la comparaison de ces données avec d'autres ressources déjà disponibles sur la polarité, ou les sentiments/émotions associés, en plus de leur exploitation dans le cadre de la désambiguïsation lexicale. De plus, l'intérêt des informations de couleur recueillies par *crowdsourcing* dans les tâches de *WSD* doit faire l'objet d'une étude complète.

Références

VON AHN, L. (2006). *Games with a purpose.* Computer, 39(6):92–94, 2006.

BERLIN, B. & KAY, P. (1969). *Basic Color Terms: Their Universality and Evolution.* Berkeley: University of California Press. ISBN 1-57586-162-3, pp. 178.

CARD, S., MACKINLAY, J. AND SHNEIDERMAN, B., (1999). *Readings in information visualization: using vision to think.* Interactive Technologies, Morgan Kaufmann, ISBN-10: 1558605339, 712 p.

CHAMBERLAIN, J., FORT, K., KRUSCHWITZ, U., LAFOURCADE, M. AND POESIO, M. (2013) *Using Games to Create Language Resources: Successes and Limitations of the Approach.* Theory and Applications of Natural Language Processing. Gurevych, Iryna; Kim, Jungi (Eds.), Springer, ISBN 978-3-642-35084-9, 2013, 42 p.

CHILD, I, L., HANSEN, J,A., AND HORNBECK, F,W. (1968). *Age and sex differences in children's color preferences.* Child Development, 39 (1):237–247.

CHRIST, R. (1975). *Review and analysis of color coding research for visual displays.* Human Factors: The Journal of the Human Factors and Ergonomics Society, 17:542–570.

GREFENSTETTE, G. (2005). *The color of things: Towards the automatic acquisition of information for a descriptive dictionary.* Revue Française de Linguistique Appliquée, 2, pp. 83–94.

JOUBERT, A;, M. LAFOURCADE, M. (2012) *A new dynamic approach for lexical networks evaluation.* In proc of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 23-25h May 2012.

KAYA, N. & EPPS, H. (2004). *Relationship between color and emotion : a study of college students.* Academic journal article from College Student Journal, Vol. 38, No. 3.

LAFOURCADE, M. (2007). *Making people play for Lexical Acquisition.* In Proc. SNLP 2007, 7th Symposium on Natural Language Processing. Pattaya, Thailande, 13-15 December 2007, 8 p.

LAFOURCADE, M. JOUBERT, A. (2012) *Increasing long tail in weighted lexical networks.* In proc of Cognitive Aspects of the Lexicon (CogAlex-III), COLING, Mumbai, India, December 2012.

LUSCHER, M. (1969). *The Luscher Color Test.* Random House, New York, New York.

MEIER, B. (1988). *Ace: a color expert system for user interface design.* In Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software, UIST '88, pages 117–128, New York, NY, USA. ACM.

NIJDAM, N A. (2010). *Mapping Emotion to Color, Human Media Interaction Human Media Interaction.* University of Twente, the Netherlands. Available at http://hmi.ewi.utwente.nl/verslagen/capita selecta/CS Nijdam Niels.pdf

OU, L-C., LUO, M.R, SUN, P-L., HU, N-C., AND CHEN, H-S. (2011). *Age effects on color emotion, preference, and harmony.* Color Research and Application.

OZBAL, G., STRAPPARAVA, C., MIHALCEA, R. AND PIGHIN, D. (2011). *A Comparison of Unsupervised Methods to Associate Colors with Words.* Affective Computing and Intelligent Interaction. Lecture Notes in Computer Science. Volume 6975, 2011, pp 42-51.

PRIBADI, N, S., WADLOW, M,G., AND BOYARSKI, D. (1990). *The use of color in computer interfaces: Preliminary research.* Information Technology Center, Carnegie Mellon University, 49 p.

SABLE, P. AND AKCAY, O. (2010). *Color: Cross cultural marketing perspectives as to what governs our response to it.* Proceedings of ASBBS, vol 17:1, -Las vegas, CA. February 2010, pages 950–954.

SAIF, M. (2011a). *Colourful language: measuring word-colour associations.* Proceeding CMCL '11. Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics. 97-106. http://www.aclweb.org/anthology-new/W/W11/W11-0611.pdf

SAIF, M. (2011b). Even *the Abstract have Colour : Consensus in Word–Colour. Associations.* In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technology. http://www.aclweb.org/anthology-new/P/P11/P11-2064.pdf

STRAPPARAVA, C., ÖZBAL, G (2010) *The color of emotions in texts.* 23rd International Conference on Computational Linguistics (COLING 2010), 28 p.

Induction de sens pour enrichir des ressources lexicales

Mohammad Nasiruddin, Didier Schwab, Andon Tchechmedjiev
Gilles Sérasset, Hervé Blanchon
Univ. Grenoble Alpes
{Mohammad.Nasiruddin, Didier.Schwab, Andon.Tchechmedjiev, Gilles.Serasset,
Hervé.Blanchon}@imag.fr

Résumé. En traitement automatique des langues, les ressources lexico-sémantiques ont été incluses dans un grand nombre d'applications. La création manuelle de telles ressources est consommatrice de temps humain et leur couverture limitée ne permet pas toujours de couvrir les besoins des applications. Ce problème est encore plus important pour les langues moins dotées que le français ou l'anglais. L'induction de sens présente dans ce cadre une piste intéressante. À partir d'un corpus de texte, il s'agit d'inférer les sens possibles pour chacun des mots qui le composent. Nous étudions dans cet article une approche basée sur une représentation vectorielle pour chaque occurrence d'un mot correspondant à ses voisins. À partir de cette représentation, construite sur un corpus en bengali, nous comparons plusieurs approches de classification non-supervisées (k-moyennes, regroupement hiérarchique et espérance-maximisation) des occurrences d'un mot pour déterminer les différents sens qu'il peut prendre. Nous comparons nos résultats au Bangla WordNet ainsi qu'à une référence établie pour l'occasion. Nous montrons que cette méthode permet de trouver des sens qui ne se trouvent pas dans le Bangla WordNet.

Abstract. In natural language processing, lexico-semantic resources are used in many applications. The manual creation of such resources is very time consuming and their limited coverage does not always satisfy the needs of applications. This problem is further exacerbated with lesser resourced languages. However, in that context, Word Sense Induction (WSI) offers an interesting avenue towards a solution. The purpose of WSI is, from a text corpus, to infer the possible senses for each word contained therein. In this paper, we study an approach based on a vectorial representation of the cooccurrence of word with their neighbours across each usage context. We first build the vectorial representation on a Bangla (also known as Bengali) corpus and then apply and compare three clustering algorithms (k-Means, Hierarchical Clustering and Expectation Maximisation) that elicit clusters corresponding to the different senses of each word as used within a corpus. We wanted to use Bangla WordNet to evaluate the clusters, however, the coverage of Bangla WordNet being restrictive compared to Princeton WordNet (23.65%), we find that the clustering algorithms induce correct senses that are not present in Bangla WordNet. Therefore we created a gold standard that we manually extended to include the senses not covered in Bangla WordNet.

Mots-clés : Induction de sens, bengali, Weka, Classification non-supervisée.

Keywords : Word Sense Induction, Bangla, Weka, Clustering.

1 Introduction

En traitement automatique des langues, les ressources lexico-sémantiques ont été incluses dans un grand nombre d'applications. La création manuelle de telles ressources est consommatrice de temps humain et leur couverture limitée ne permet pas toujours de couvrir les besoins des applications. Ce problème est encore plus important pour les langues moins dotées que le français ou l'anglais. L'induction de sens présente dans ce cadre une intéressante piste. À partir d'un corpus de texte, il s'agit d'inférer les sens possibles pour chacun des mots qui le composent. Nous étudions dans cet article une approche basée sur une représentation vectorielle pour chaque occurrence d'un mot correspondant à ses voisins. À partir de cette représentation, construite sur un corpus en bengali, nous comparons plusieurs approches de classification non-supervisée des occurrences d'un mot pour déterminer les différents sens qu'il peut prendre. Nous montrons que cette méthode permet de trouver des sens

qui ne se trouvent pas dans le *Bangla WordNet*.

Dans cet article nous commençons par présenter le bengali, le *Bangla WordNet* et le Wikipédia bengali qui constitue notre corpus. Nous présentons ensuite l'approche qui nous a permis de construire la représentation vectorielle pour chacune des occurrences de notre corpus. Enfin nous présentons les deux méthodes d'évaluation que nous utiliserons pour comparer les trois différents algorithmes de classification non-supervisée.

2 Le bengali et Bangla WordNet

2.1 Le bengali

Le bengali, également appelé bangla, est la septième langue la plus parlée au monde avec environ 200 millions de locuteurs et la plus orientale des langues indo-européennes. Elle est essentiellement parlée au Bangladesh (75% de la population) et la deuxième la plus parlée en Inde où elle est langue officielle de trois des vingt trois états (Garry & Rubino, 2001). Le bengali est écrit à l'aide de caractères dérivant du Brahmi. Comme le français, les mots bengalis sont séparés par des espaces et les signes de ponctuation sont les mêmes à part le dari (।) qui remplace le point pour la segmentation des phrases.

2.2 Bangla WordNet

Bangla WordNet (Dash, 2011 ; Niladri Sekhar Dash & Banerjee, 2011) a été conçu en suivant les principes du WordNet de Princeton pour l'anglais. Il fait partie du projet Indradhanush et a ainsi été développé en utilisant Hindi WordNet (Somesh Jha & Bhattacharyya, 2010) comme pivot. Classiquement, un synset du *Bangla WordNet* est composée ainsi :

- un numéro d'identification du synset : un numéro unique provenant de l'Hindi WordNet.
- une catégorie grammaticale : nom, verbe, adjectif, adverbe.
- une glose : la description du concept par une définition et des exemples.
- un ensemble de mots, synonymes entre eux, dont les traits communs sont sensés correspondre au sens particulier décrit pas le synset.

Par exemple, le synset associé à সমাগত (samāgata – arrivé) est :

ID	:: 7	
CAT	:: ADJECTIVE	
CONCEPT	:: যে এসেছে	(Il est arrivé.)
EXAMPLE	:: "সমাগত ব্যক্তিদের স্বাগত জানাও"	(Les nouveaux arrivants sont les bienvenus.)
SYNSET-BENGALI	:: আগত, সমাগত	(imminemment, arrivé)

Les traductions en français ne se trouvent pas dans le *Bangla WordNet* et sont données uniquement pour faciliter la compréhension des lecteurs.

Parties du discours	Synsets	Nombre de sens	Nombre de sens (mots simples)	Nombre de sens (mots composés)	Mots monosémiques	Mots polysémiques
Nom	27 281	44 854	34 496	10 358	29 760	5 829
Verbe	2 804	4 448	318	4 130	2 260	671
Adjectif	5 815	10 264	569	9 695	6 813	1 385
Adverbe	445	906	159	747	721	79
Total	36 345	60 472	35 542	24 930	39 554	7 964

Table 1 – Statistiques sur le *Bangla WordNet*

2.3 Wikipédia bengali

Alors que le bengali est la septième langue la plus parlée au monde, le Wikipédia bengali [1] n'est que le quatre-vingt cinquième en nombre d'articles. Au 1er mai 2014 à midi-UTC il comprenait 29 756 articles contre, par

1. https://bn.wikipedia.org

exemple, 106 097 articles pour le latin ou 178 760 articles pour le basque, deux langues bien moins parlées au quotidien que le bengali [2]. Le faible nombre d'articles en bengali est évidemment explicable par le niveau d'éducation dans les régions où il est parlé. Par exemple, au Bangladesh et dans l'état du Bengale-Occidental qui représentent à eux deux trois-quarts des locuteurs du bengali, le taux de scolarisation dans le secondaire est inférieurs à 50% selon l'UNICEF [3] et l'OCDE (OCDE, 2011).

Nous allons utiliser le Wikipédia bengali comme corpus de textes dans notre expérience. Utiliser un Wikipédia est, en effet, un moyen simple d'obtenir des textes libres de droits pour toutes les langues où il en existe un. Dans cette expérience, nous utilisons la sauvegarde de la base de données du Wikipédia bengali [4] du 28 décembre 2013 [5] qui comporte 28 393 articles.

3 Construction de la matrice des voisinages

Dans cette section, nous présentons l'expérience réalisée qui a consisté à construire la matrice des voisinages puis à catégoriser chacune des instances de mots en fonction de leur contexte.

Nous appelons *occurrence* une suite de caractères délimitée par des séparateurs (espace, virgule, dari, *etc.*). Nous appelons *forme*, l'ensemble de toutes les occurrences ayant en commun leur suite de caractères. Nous appelons S l'ensemble des formes du corpus.

Si on considère le corpus constitué des deux phrases suivantes, "*le chat mange la souris*" "*le chat mange le fromage*", il est composé de 10 occurences et S contient 6 formes. L'élément ‹*chat*› de S a deux occurences que nous noterons ‹*chat#1*› et ‹*chat#2*›.

3.1 Préparation des données

Le texte brut des pages du Wikipédia bengali a été extrait et normalisé selon la forme normale NFD (*Normalization Form Canonical Decomposition*). Chaque phrase des pages Wikipédia est considérée comme un document indépendant. Nous obtenons donc un ensemble de 34 251 documents correspondant aux 28 393 pages originales. Enfin, chaque document est segmenté en une séquence d'instances.

Le Bangla ne distingue pas les majuscules des minuscules et a une morphologie relativement peu productive. Ainsi, aucun autre pré-traitement linguistique (lemmatisation, annotations en partie du discours, *etc.*) n'a été appliqué au corpus.

3.2 Mise en œuvre

Après le pré-traitement du corpus, nous construisons la matrice formes-contextes et nous comparons plusieurs algorithmes de classification pour la création des sens.

3.2.1 Construction de la matrice des voisinages

Dans cette matrice, les lignes correspondent aux éléments de S et les colonnes correspondent à leurs occurrences. Chaque case de la matrice contient le nombre de fois où ces occurrences se trouvent dans le même document que l'élément. Par exemple, si on considère que notre corpus n'est composé que des deux documents suivants, "*le chat mange la souris*" "*le chat mange le fromage*", la matrice résultat sera celle présentée dans la table 2. Notre corpus fait 34 251 documents qui contiennent 3 957 075 instances et 373 685 formes. Nous avons utilisé une machine de 32 cœurs Intel Xeon E5-2650 à 2.0 GHz équipée de 256 Go de ram. La génération de cette matrice a pris environ 48 heures mais n'a utilisé aucune parallélisation.

2. https://meta.wikimedia.org/wiki/List_of_Wikipedias
3. http://www.unicef.org/french/infobycountry/bangladesh_bangladesh_statistics.html
4. http://dumps.wikimedia.org/backup-index-bydb.html
5. http://dumps.wikimedia.org/bnwiki/20131228/

	le#1	chat#1	mange#1	la#1	souris#1	le#2	chat#2	mange#2	le#3	fromage#1
le	0	1	1	1	1	1	2	2	1	2
chat	1	0	1	1	1	1	0	1	1	1
mange	1	1	0	1	1	1	1	0	1	1
la	1	1	1	0	1	0	0	0	0	0
souris	1	1	1	1	0	0	0	0	0	0
fromage	0	0	0	0	0	1	1	1	1	0

Table 2 – Matrice obtenue avec l'exemple

3.2.2 Construction des groupes

L'objectif de ce travail est de voir dans quelle mesure il est possible d'inférer des sens à partir de groupes pour créer une ressource lexicale. Nous voulons ainsi comparer plusieurs algorithmes de classification pour savoir lequel ou lesquels seraient les plus performants pour cette tâche. Nous avons ainsi utilisé la suite de logiciels d'apprentissage automatique Weka (*Waikato Environment for Knowledge Analysis*) pour créer les groupes de sens. Nous expérimentons ici trois algorithmes : k-moyennes, regroupement hiérarchique et espérance-maximisation, qui utilisent une distance euclidienne.

k-moyennes (Hartigan & Wong, 1979) est un algorithme numérique, non-supervisé, probabiliste et itératif de partition de données. Il permet de générer un nombre de groupes k donné en paramètre. Aucune occurences ne peut appartenir à deux groupes à la fois.

Le regroupement hiérarchique (Johnson, 1967) est un algorithme qui unit les groupes les plus proches jusqu'à ce que le nombre de groupes voulu soit atteint.

L'algorithme d'espérance-maximisation (EM) (Jin & Han, 2010) estime par maximum de vraisemblance les paramètres d'un modèle probabiliste ayant des variables latentes. Une gaussienne de paramètre inconnu modélise l'ensemble des points d'un groupe. Une distribution modélise la vraisemblance d'appartenance des points aux groupes. EM estime conjointement les paramètres des gaussiennes afin de maximiser la vraisemblance d'appartenance aux groupes. Contrairement à k-moyennes et le regroupement hiérarchique, EM travaille sur les distributions des points, ce qui est une approche orthogonale qui peut amener des résultats complémentaires potentiellement meilleurs.

4 Évaluation des catégories

4.1 Le Bangla WordNet et ses sens dans le corpus

Comparé au Princeton WordNet (Miller, 1995), le *Bangla WordNet* (BWN) a une couverture de seulement 23,65% (Dash, 2011). Nous avons voulu essayer d'évaluer sur un petit sous-ensemble des entrées combien de sens manquaient. Nous avons choisi 7 mots au hasard et un natif bengali a annoté les occurrences de ces mots dans les 258 documents et 14 817 phrases dans lesquelles ils apparaissent. Si le sens existe dans le *Bangla WordNet*, il a annoté avec ce sens sinon il a créé de nouveaux sens. Cette annotation lui a pris environ 8 heures.

La table 3 présente les résultats obtenus avec ces 7 mots. On le voit, pour seulement sept mots, avec un corpus assez simple, on trouve 20 sens manquants à *Bangla WordNet* soit au moins 57% des sens possibles pour ces mots. L'enrichissement de cette ressource est donc un objectif assez primordial. Nous étudions dans la partie suivante dans quelle mesure nous pouvons le faire toujours en nous basant sur nos 7 mots.

4.2 Évaluation

L'évaluation se fait par rapport à deux références : le *Bangla WordNet* et le *Bangla WordNet* enrichi par les sens manquants découverts par l'annotateur. L'élément le plus important pour l'évaluation est le choix de mesures de la qualité de la classification par rapport aux références.

Mots	Nombre d'occurrences	Sens *Bangla WordNet*	Sens manquant	Total BWN + corpus
সমাগত (samāgata – arrivé)	8	2	0	2
অংক (aṅka – math)	26	2	4	6
অচল (acala – immobile)	40	5	1	6
পদবী (padabī – titre)	113	1	6	7
জনপদ (janapada – communauté)	83	1	3	4
যুক্তাক্ষর (yuk-tākṣara – lettre composée)	12	1	0	1
অটল (aṭala – régulier)	43	3	6	9
Total	325	15	20	35

Table 3 – Statistiques sur sept mots pris au hasard dans le corpus

4.2.1 Mesures

Nous utilisons 4 mesures standard pour l'évaluation de la qualité de la classification. Le score de chaque groupe est calculé comme la somme de la mesure entre le groupe et chacun des groupes de référence. Sur l'ensemble des groupes, on donne la moyenne des scores pour chaque groupe.

- La F1-mesure est la moyenne harmonique entre la précision (P) et le rappel (R) et s'exprime comme $F1 = \frac{2 \cdot P \cdot R}{P+R}$. P est le nombre de points correctement assignés (vrais positifs, VP) sur le nombre total de points (somme de VP + les faux positifs, FP). R est le nombre de points correctement assignés sur le nombre de points attendus (somme de VP et des faux négatifs, FN).
- L'indice de Jaccard sert à calculer la similarité entre un groupe et un groupe de référence en comptant le nombre d'éléments communs sur le nombre total d'éléments des deux groupes. $JI(C, C_{ref}) = \frac{TP}{VP+FP+FN}$
- L'indice de Rand permet de calculer le pourcentage d'éléments assignés correctement, $RI(C, C_{ref}) = (VP + VN)/(VP + FP + FN + VN)$.
- L'indice de Rand ajusté est un indice Rand qui est ajusté pour prendre en compte l'assignation correcte ou incorrecte due au hasard en prenant en compte la distribution de l'ensemble des indices de rand sur les groupes : $ARI(C, C_{ref}) = \frac{Indice - IndiceEspr}{IndiceMaximum - IndiceEspr}$

4.2.2 Résultats

La table 4 présente les résultats de trois algorithmes k-moyennes, regroupement hiérarchique et EM. À droite, la référence est le *Bangla WordNet* tandis qu'à gauche, la référence est le standard que nous avons créé. Pour

Algorithmes	F1	JI	RI	ARI
k-moyennes	**54.91**	42.85	42.85	2.36
Hiérarchique	46.06	**61.34**	**55.41**	**4.78**
EM	49.19	39.73	39.98	2.30

Algorithmes	F1	JI	RI	ARI
k-moyennes	50.69	36.62	40.69	3.16
Hiérarchique	**52.05**	**58.09**	**62.54**	**5.80**
EM	44.21	34.40	43.42	2.63

Table 4 – Résultats obtenus avec la référence *Bangla WordNet* (à gauche) avec la référence créée (à droite)

la référence *Bangla WordNet*, si l'on s'en tient au F1-score, k-moyennes obtient la meilleure performance par rapport à EM et au regroupement hiérarchique (résp. +8,85, +4,72). En revanche, avec les trois autres mesures c'est le regroupement hiérarchique qui est de loin le meilleur (par rapport au second meilleur, l'indice de Jaccard (JI) — +18,49, l'indice de Rand (RI) — +12,56, l'indice de Rand ajusté (ARI) — +2,52). k-moyennes et EM sont proches, mais k-moyennes reste toujours devant. Pour la référence que nous avons créé, nous constatons que pour toutes les mesures, le regroupement hiérarchique est meilleur (par rapport au second meilleur, F1-score — +1.36, l'indice de Jaccard — +21,47, l'indice de Rand — +19,12, l'indice de Rand ajusté — +2,64). k-moyennes est deuxième pour toutes les mesures, sauf pour l'indice de Rand. En fonction des applications et de l'importance des vrais et faux négatifs, certaines mesures seront plus représentatives que d'autres. L'indice de Rand ajusté est dans le cas général considéré comme la mesure la plus représentative.

5 Conclusions et perspectives

Dans cet article, nous avons présenté une première approche d'enrichissement d'une base lexico-sémantique en particulier pour des langues moins dotées que l'anglais comme l'est le bengali pourtant septième langue la plus parlée au monde. Par l'étude des occurrences de sept mots dans un corpus constitués des textes du Wikipédia du bengali, nous avons montré que 20 sens n'étaient pas répertoriés dans le *Bangla WordNet* contre 15 qui s'y trouvaient (soit un taux d'absents de 57%). La mise en œuvre d'une méthode très simple nous a permis de découvrir automatiquement des sens qui ne se trouvaient pas dans la ressource initiale. Nos travaux actuels visent à améliorer la construction des groupes en particulier en exploitant les informations issues de la ressource initiale et à les évaluer *in vivo*, c'est-à-dire dans une application comme la traduction automatique.

Remerciements

Nous tenons à remercier chaleureusement Dr. Niladri Sekhar Das et Prof. Dr. Pushpak Bhattacharyya d'avoir gracieusement accepté de mettre *Bangla WordNet* à notre disposition.

Références

Dash N. S. (2011). Problems in defining language specific synsets (lss) in bengali for the indradhanush indo-wordnet : Some theoretical and practical issues. In *Proceedings of the 2nd National Workshop of Indradhanush WordNet Consortium*, p. 4–18.

Garry J. & Rubino C. (2001). Facts about the world's languages. *HW Wilson*.

Hartigan J. A. & Wong M. A. (1979). Algorithm as 136 : A k-means clustering algorithm. *Applied statistics*, p. 100–108.

Jin X. & Han J. (2010). Expectation maximization clustering. In *Encyclopedia of Machine Learning*, p. 382–383. Springer.

Johnson S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, **32**(3), 241–254.

Miller G. A. (1995). Wordnet : a lexical database for english. *Communications of the ACM*, **38**(11), 39–41.

Niladri Sekhar Dash, Abhisek Sarkar D. B. & Banerjee S. (2011). Problems and challenges in translation of hindi synsets into bengali in indradhanush wordnet. In *Proceedings of the 2nd National Workshop of Indradhanush WordNet Consortium*, p. 19–38.

OCDE (2011). *Études économiques de l'OCDE : Inde 2011*. Rapport interne, OCDE.

Somesh Jha, Darren Narayan P. P. & Bhattacharyya P. (2010). A wordnet for hindi. In *Proceedings of the International Workshop on Lexical Resources in Natural Language Processing*.

Un dictionnaire et une grammaire de composés français

François Trouilleux
Clermont Université, Université Blaise-Pascal, EA 999, LRL
francois.trouilleux@univ-bpclermont.fr

Résumé. L'article présente deux ressources pour le TAL, distribuées sous licence GPL : un dictionnaire de mots composés français et une grammaire NooJ spécifiant un sous-ensemble des schémas de composés.

Abstract. The paper introduces two resources for NLP, available with a GPL license: a dictionary of French compound words and a NooJ grammar which specifies a subset of compound patterns.

Mots-clés : open source, ressources, dictionnaire, grammaire, mots composés
Keywords: open source, resources, dictionary, grammar, compound words

1 Introduction

Le logiciel NooJ (Silberztein, 2003) dispose d'un dictionnaire en source ouverte pour le français. Le présent article propose de le compléter par un dictionnaire de mots composés, qu'on appelle ici DM-C. Plusieurs dictionnaires électroniques utilisables pour le TAL sont accessibles gratuitement. Nous en avons observé cinq : le DELA (Silberztein, 1990), le Lefff (Sagot, 2010), Morphalou (Romary et al., 2004), Lexique (New, 2006) et Multext (Véronis, 1998). D'autres ressources électroniques pourraient être prises en compte, notamment le Wiktionnaire ou (Dubois et Dubois-Charlier, 2001), mais leur format (Wiki ou PDF) est moins directement exploitable [1]. M. Mathieu-Colas (1995a) a développé un dictionnaire formalisé contenant 13 000 formes à trait d'union, mais qui n'est, semble-t-il, pas diffusé.

Le tableau 1 ci-dessous permet d'introduire à la fois la problématique de ce travail et la solution que nous présentons ; il donne une comparaison quantitative des cinq dictionnaires considérés, pour les formes à trait d'union [2]. Les deux premières colonnes donnent le nombre de lemmes et de formes dans chaque dictionnaire. La colonne suivante (t. l.) donne le ratio nombre de lemmes / nombre de formes (« taux de lemmatisation »). Morphalou ne lemmatise pas du tout les composés ; le plus souvent, seule la forme du singulier ou l'infinitif figure dans le dictionnaire et quand plusieurs formes fléchies figurent dans le dictionnaire, chacune est son propre lemme. Le Lefff et Multext lemmatisent le mieux ; verbes mis à part, le taux de lemmatisation est de 1,76 pour ces deux dictionnaires. Le DELA et Lexique contiennent des verbes mais ne les fléchissent que très incomplètement.

Les six colonnes suivantes donnent pour chaque dictionnaire (par ligne) la part du total (en % du nombre de formes) qui est contenue dans chacun des autres (par colonne). Les chiffres de ces colonnes révèlent un taux de recouvrement assez faible entre les dictionnaires. Pour les compléter, notons que l'union des cinq dictionnaires compte 13 617 formes, dont seulement 4 344 (32 %) se retrouvent dans au moins deux dictionnaires et 569 (13 %) se retrouvent dans les cinq. Le DM-C intègre de 61 à 90 % des formes des cinq dictionnaires que nous avons observés. Pour des raisons qui apparaitront plus loin, le choix a été fait de ne pas viser un dictionnaire qui serait l'union des cinq dictionnaires de départ. Au DM-C s'ajoute une grammaire permettant de reconnaître différents types de composés. Pour chaque dictionnaire, la colonne G indique combien de mots de ce dictionnaire ne figurant pas dans le DM-C sont reconnus par la grammaire (en % du nombre de formes du dictionnaire). La colonne *total* fait la somme des deux colonnes

[1] Le Wiktionnaire a fait l'objet d'une conversion dans un format exploitable pour le TAL (GLAFF), mais les composés ne sont pas traités pour l'instant, cf. (Sajous *et al*., 2013).

[2] Par « forme », on entend le mot lui-même indépendamment de sa catégorie ou des traits qui lui sont associés. Les formes contenant un espace et un trait d'union (ex. *à rebrousse-poil*) ne sont pas comptées dans ce tableau (le DELA contient un nombre très important de composés avec espace, qui rendrait les croisements illisibles).

précédentes et indique donc le taux de couverture du couple DM-C + grammaire pour chaque dictionnaire. On obtient 99 % pour Morphalou car ce dictionnaire a servi de ressource centrale pour le DM-C. Le 1 % restant représente 46 formes volontairement exclues (essentiellement des erreurs).

	lemmes	formes	t. l.	dela	lefff	lex.	mph	mult.	dm-c	G	total
dela	3855	4997	1,3	100	29	33	29	29	61	24	85
lefff	1772	3307	1,87	44	100	49	39	58	89	6	95
lex.	3430	4243	1,24	38	39	100	35	42	68	25	93
mph	5608	5608	1	26	23	26	100	21	64	35	99
mult.	1658	4072	2,46	36	47	44	29	100	90	8	98

TABLE 1. Comparaison quantitative des dictionnaires pour les mots avec trait d'union

L'alinéa suivant présente des observations sur le contenu des cinq dictionnaires que nous avons retenus, qui justifient le développement de nos deux ressources. L'alinéa 3 présente notre grammaire NooJ pour un sous-ensemble de schémas de composition, et l'alinéa 4 notre dictionnaire et les solutions qu'il apporte aux problèmes observés.

2 Observation des dictionnaires

Le codage des composés en français fait l'objet de fluctuations bien connues, dont on retrouve la trace dans les dictionnaires électroniques. Le tableau 2 donne des exemples de formes différentes trouvées dans les cinq dictionnaires. Un premier axe de variation concerne l'usage du **trait d'union** *vs* **soudure** ou **espace**. N. Catach (1981) a montré que des dictionnaires imprimés variaient sensiblement sur ce point ; cette variation se retrouve dans les dictionnaires électroniques. Les exemples donnés dans les premières lignes du tableau 2 parlent d'eux-mêmes.

On sait aussi par ailleurs que les formes du **singulier** ou du **pluriel** de certains noms composés sont imprécises. Les rectifications de l'orthographe proposées en 1990 ont consacré cette variabilité en autorisant officiellement les variantes (cf. Catach, 1991). Le traitement des mots *accroche-cœur* et *monte-plats* illustre la variabilité à l'œuvre dans les dictionnaires (*s* et *p* indiquent le nombre associé, – indique l'absence de trait de nombre). Les noms *année-lumière* et *mot-valise* illustrent en outre des problèmes de couverture et de précision. Il est à noter que l'absence de marque de genre dans Morphalou ne signifie pas une invariabilité, mais plutôt une absence d'information sur les formes fléchies.

Un troisième axe de variation, moins souvent évoqué car plus spécifique aux dictionnaires pour le TAL, concerne la **segmentation** des unités lexicales. Si la lecture des entrées *plain-pied* et *stand-by* dans le TLF suggère que ces mots soient considérés comme des noms, l'expression *bras-le-corps* semble bien impossible sans la préposition *à*, ce qui doit conduire à ne retenir qu'une entrée adverbiale *à bras-le-corps*.

Enfin, on rencontre aussi des **variantes orthographiques**, telles que *laisser-passer* ou *medicine-ball*. En règle générale, ces variantes ne sont pas signalées par l'attribution d'un lemme commun aux deux formes, si bien que le lemme *laisser-passer* est compté comme figurant dans deux dictionnaires, alors que le concept, si l'on peut dire, est connu des cinq.

Les disparités relevées en terme de recouvrement des dictionnaires peuvent s'expliquer aussi par le fait que la composition est un procédé **productif**. Par exemple, Lexique contient *en propre* 88 lemmes associés à des formes avec un préfixe *re-* (p.ex. *re-contacter*, *re-belote*, mais aussi *re-café*, *re-drapeau*, *re-main*...), 163 avec *ex-*, 22 avec *co-*. Morphalou contient 123 formes commençant par *anti-* dont seulement 11 se retrouvent dans un autre dictionnaire. Le TLF, dont est dérivé Morphalou, donne des composés parfois comme mot vedette (p.ex. *presse-citron*), parfois à l'intérieur des articles (p.ex. *pubo-fémoral* figure dans l'article *fémoral*). En particulier, le TLF a dans sa nomenclature des préfixes, suffixes et autres « éléments formants », pour lesquels les articles donnent des listes de composés. Seulement 39,7 % des mots à trait d'union de Morphalou figurent comme mots vedettes dans le TLF.

L'objectif d'un dictionnaire classique comme le TLF est multiple : fournir un relevé des mots attestés, en donner la définition, en expliquer les emplois. S'agissant des composés, le premier de ces objectifs n'est pas central pour un dictionnaire destiné au TAL. Il est au moins aussi important d'avoir un système rendant compte des mécanismes de composition, qui permettra de détecter les nouvelles productions. On est tiraillé dans deux directions opposées : la lexicalisation d'un maximum d'unités et la reconnaissance des unités composées par des règles. C'est la raison pour laquelle le traitement que nous proposons se fait en partie par une grammaire, en partie par le dictionnaire.

DELA	***Lefff***	***Lexique***	***Morphalou***	***Multext***
contreplongée	*contre-plongée*	*contre-plongée*	*contreplongée*	
sociolinguistique	*sociolinguistique*		*sociolinguistique* *socio-linguistique*	*sociolinguistique*
cul de basse fosse		*cul-de-basse-fosse*	*cul de basse-fosse*	*cul-de-basse-fosse*
		pan bagnat		*pan-bagnat*
accroche-cœurs s *accroche-cœurs* p	*accroche-cœur* –	*accroche-cœur* s *accroche-cœurs* p	*accroche-cœur* –	*accroche-cœur* s *accroche-cœur* p *accroche-cœurs* p
monte-plats s *monte-plats* p	*monte-plat* p *monte-plats* –	*monte-plat* s *monte-plats* –	*monte-plats* –	*monte-plat* s *monte-plats* p
années-lumière p	*année-lumière* s *années-lumières* p	*année-lumière* s *années-lumière* p		
mot valise s	*mot-valise* s *mot-valises* p	*mot-valise* s	*mot-valise* –	
stand-by,N	*en stand-by*,ADV	*stand-by*,N	*stand-by*,N	*stand-by*,N
de plain pied,ADV *de plain-pied*,A		*plain-pied*,N	*plain-pied*,ADV *plain-pied*,N	*de plain-pied*,ADV
à bras-le-corps,ADV	*bras-le-corps,N*		*à bras-le-corps,ADV* *bras-le-corps,ADV*	
laissez-passer *laisser-passer*	*laissez-passer*	*laissez-passer*	*laissez-passer* *laisser-passer*	*laissez-passer*
		médecine-ball *medicine-ball*	*médecine-ball*	*médecine-ball* *medicine-ball*

TABLE 2. Exemples de disparités entre dictionnaires.

3 Une grammaire pour un sous-ensemble des composés

Définir un jeu de règles décrivant des schémas productifs de composition peut sembler a priori sans difficulté. M. Mathieu-Colas (1996), par exemple, donne une liste très détaillée de tels schémas. La difficulté apparaît cependant quand il s'agit de détecter les composés dans les textes *sans générer d'analyses incorrectes*. Nous présentons ici une grammaire qui porte sur un sous-ensemble de composés qui peuvent être identifiés syntaxiquement, hors contexte, avec une très haute précision, ce qui rend leur inclusion dans un dictionnaire dispensable.

Si on tente d'identifier les composés en présence d'un trait d'union, qu'on peut voir comme leur signature, on s'expose à deux problèmes : celui des juxtapositions (ex. *l'opposition consonnes-voyelles*) et surcompositions (ex. *un porte-filtre à café*), cf. (Mathieu-Colas, 1995b) et celui des ambiguïtés lexicales (p.ex. *ferme-auberge* ne désigne ni une auberge qui aurait une certaine fermeté, ni un instrument pour fermer les auberges). Etant donné ces difficultés, la lexicalisation des composés reste la meilleure solution pour les schémas N-N, N-A, A-A et V-N. En revanche, on peut identifier un sous-ensemble de schémas qui peuvent faire l'objet d'une analyse quasi déterministe. C'est ce que nous avons fait dans une grammaire NooJ, dont le tableau 3 donne les caractéristiques.

Quelques préfixes et quelques-uns des 53 verbes listés comme productifs pour le schéma V-N sont aussi des noms et attestés en position 1 d'un composé N-N ou N-A, comme dans p.ex. *auto-école, micro-cravate, photo-finish, serre-tunnel*... On a pris soin d'inclure ces suites N-N ou N-A dans le DM-C et on a inclus dans la grammaire, outre les schémas du tableau 3, deux règles permettant de construire ce type de suites avec ces formes ambigües. La grammaire reconnait donc par exemple *télé-surveillance* par deux schémas : PFX-N ou N-N.

Le schéma le plus productif dans cet ensemble, relativement aux dictionnaires, est de loin la composition avec préfixe (cf. tableau 4). D'autres schémas s'y apparentent : mot fonctionnel introducteur de syntagme (préposition ou numéral) ou modifieur antéposé (adjectif, adverbe ou point cardinal). La lexicalisation de 53 formes verbales permet en quelque sorte de transformer ces verbes en « préfixes », à la manière de la catégorie « élément de composition » du TLF. On rejoint donc dans cette grammaire la stratégie du *chunking*, qui cible les séquences précédant les têtes de syntagmes. Les schémas V-GP et N-GP relève quant à eux d'une stratégie de reconnaissance de la plus longue chaine : en présence d'un composé comme *Vaires-sur-Marne*, en corpus, il faut éviter de reconnaitre *sur-Marne* comme un composé PFX-N.

Schémas	***Exemples***
Composés sur préposition. Avec les prépositions *à*, *après*, *avant*, *en*, *hors*, *outre*, *pour* ou *sans*.	*à-côté, avant-programme, en-but, sans-opinon*
Composés avec préfixes. Les préfixes se combinent avec des mots de catégorie ADV, A, N ou V pour donner un mot de même catégorie. Le DM a été développé pour ce projet et contient 1371 préfixes.	*quasi-contractuellement, turbo-train, extra-plat, artério-veineux, sous-traiter, pré-enregistrer*
Composés sur verbe. Deux infinitifs, ou un verbe avec un adverbe antéposé, ou encore le schéma très productif V-N mais limité à un ensemble de 53 verbes identifiés comme les plus productifs [3].	*savoir-faire, bien-être, bien-aimé, attrape-couillon, cache-col, lave-linge, porte-bébé*
Composés numéral – nom pluriel.	*quatre-chevaux, trois-pièces*
Composés adjectif – nom. Schéma limité à 19 adjectifs attestés avec au moins trois occurrences dans un composé A-N du DM-C ou 6 autres adjectifs attestés et apparentés par le sens à l'un des 19 [4].	*blanc-seing, court-bouillon, double-toit, morte-saison, rouge-gorge, vif-argent*
Composés *sud*, *nord*, *est* ou *ouest* – adjectif.	*ouest-allemand, sud-vietnamien*
Composés verbe ou nom – groupe prépositionnel (V-GP et N-GP).	*tape-à-l'œil, abri-sous-roche*

TABLE 3. Types de composés identifiés par la grammaire.

4 Dictionnaire

Le DM-C a été construit en prenant Morphalou comme noyau central. Il contient en premier lieu les formes à trait d'union appartenant à au moins deux des cinq dictionnaires que nous avons examinés ou qui appartiennent à Morphalou seulement mais ne sont pas reconnues par notre grammaire. Le premier de ces deux critères permet de filtrer des erreurs. Le second critère permet de s'appuyer sur une ressource dont la licence permet clairement le réemploi et, pour les vérifications, sur les définitions du *Trésor de la langue française*, dont est extrait Morphalou. Le DM-C intègre également les lemmes composés avec espace de Morphalou (un peu moins de 300 lemmes), et les mots fonctionnels et quelques adverbes figurant déjà dans le DM des mots simples. On obtient un dictionnaire de 4910 lemmes.

4.1 Flexion des composés

On a vu dans l'introduction (cf. tableau TABLE 1) que Morphalou ne lemmatise pas les mots à trait d'union et contient peu de formes fléchies. C'est un des apports du DM-C que de fournir une flexion systématique des composés qu'il contient. Le formalisme de NooJ permet de définir des modèles de flexion auxquels on associe ensuite les entrées du dictionnaire. On a défini pour le DM-C 74 nouveaux modèles de flexion, qui s'ajoutent à l'ensemble déjà défini pour le DM.

Le pluriel des composés de type V-N et PREP-N a fait l'objet d'une réforme en 1990. Le DM-C intègre les recommandations de cette réforme, qui, rappelons-le, préconise des formes qui *s'ajoutent* à la graphie traditionnelle. Ainsi, les mots *accroche-cœur*, *monte-plat* et *taille-crayon* sont associés respectivement aux trois modèles de flexion qui suivent :

```
M_S_0     = <E>/m+s      |  s/m+p+Rec  |  <E>/m+p+Opt ;
M_0_S     = <E>/m+s+Rec  |  s/m+s+Opt  |    s/m+p     ;
M_0_S_0_S = <E>/m+s+Rec  |  s/m+s+Opt  |  <E>/m+p+Opt  |  s/m+p+Rec ;
```

Ces modèles génèrent deux pluriels pour *accroche-cœur*, deux singuliers pour *monte-plat* et deux singuliers, deux pluriels pour *taille-crayon*. 296 + 86 + 43 (78 %) des 544 composés de type V-N sont associés à ces trois modèles de flexion. Suivant une notation adoptée depuis la version 1.3 du DM, le trait `Rec` marque la graphie recommandée par la réforme et le trait `Opt` (« optionnel ») la graphie archaïsante. Le DM-C est à notre connaissance le premier dictionnaire pour le TAL à intégrer de façon systématique la flexion nouvelle des composés.

[3] Verbes bases d'au moins trois composés de type V-N, V-PRO (*brûle-tout*) ou V-GN (*trompe-la-mort*) dans Morphalou.

[4] Au moins 3 occurrences: *bas, beau, blanc, bon, court, double, faux, franc, grand, gros, haut, libre, mort, nu, petit, plat, plein, saint, tiers* ; apparentés : *long, rouge, vif, prime, quart, vert.*

4.2 Origine des composés et schémas de composition

Outre une flexion systématique, le DM-C donne deux informations supplémentaires : l'origine du mot ou son schéma de composition. Le DM-C contient 482 composés d'origine étrangère. Ces mots ont un attribut `Lang` dont la valeur est le code ISO 639 de la langue d'origine. La langue la plus fréquente est l'anglais : 58 % des mots d'origine étrangère, suivie du latin (30 %), de l'italien (5 %), puis de 19 langues différentes, chacune avec moins de 5 mots.

Les noms, verbes et adjectifs proprement français ont un attribut `Cmp` dont la valeur indique le schéma de composition. Le tableau 4 donne les fréquences de 43 schémas dont le code est obtenu par croisement des en-têtes de ligne et de colonnes. Ainsi, par exemple, 800 lemmes sont construits sur un schéma préfixe-nom (code PFX_N). A ces 43 codes s'ajoutent quelques autres codes, dont dCMP (dérivé de composé, ex. *court-circuiter*, 128 occ.), REP et QREP (répétition ou quasi-répétition, ex. *fric-frac*, 57 occ.), PH (composés par phrase, ex. *m'as-tu-vu*, 23 occ.).

	N	A	GP	PP	G	V	PRO	ADV	GN	SFX	NUM	P	total
PFX	800	513				35	1			10	1		1360
N	648	88	176	5	2			1	2				922
A	256	25	11	8	1	1							302
ADV	1	5	1	15	3	10						1	36
V	545	15	19			14	11	11	13		1		629
P	88	3				2	11		7		2		113
NUM	25		1								3		29
total	2363	649	208	28	6	62	23	12	22	10	7	1	3391

Table 4. Fréquence des types de composés.

4.3 Gestion des variantes

L'usage du trait d'union, comme on l'a rappelé au §2, est assez flottant. Le TLF contient des entrées dont la forme graphique inclut un trait d'union facultatif, telles que *saute(-)en(-)barque* ou *auto(-)féconder*. L'ambiguïté d'expansion des parenthèses – orthographe sans ou avec espace possible – a conduit les concepteurs de Morphalou à retenir seulement la version avec trait d'union. Morphalou a donc un biais en faveur des graphies avec trait d'union. On adopte pour le DM-C une approche qui tolère assez largement les alternances soudure/trait d'union et trait d'union/espace, grâce à deux caractères spéciaux fournis par NooJ : _ et =, qui sont interprétés respectivement comme « la chaine vide, un trait d'union ou un espace » et « un trait d'union ou un espace ». Ainsi, par exemple, une entrée `week_end,N` permet de reconnaitre à la fois *weekend*, *week-end* et *week end*.

Sont déclarés dans le DM-C avec le caractère _ les mots dont la soudure est recommandée par la réforme de 1990 (ex. *hotdog*) et les composés sur préfixe, à l'exception des préfixes *sous*, *demi*, *mi*, *semi*, *self*, *vice*, *ex* (signifiant « antérieurement »), des préfixes référant à des peuples (*franco*, *judéo*...) et des formes *non*, *quasi*, *arrière* et *social* que le DM catégorise aussi comme « préfixes ». On tient aussi compte d'exceptions telles que *extra-utérin* ou *contrexpertise*.

Sont déclarés avec le caractère =, de façon systématique, les noms composés de type N-A (*garde=champêtre*, *amour=propre*), N-N (*allocation=chômage*), N-GP (*bec=de=lièvre*) et V-N (*vide=grenier*), les cardinaux et ordinaux avec *et* (*vingt=et=unième*) et les mots étrangers (quand ils n'admettent pas aussi la soudure ; *pater=familias*, *pan=bagnat*). S'y ajoutent certains noms A-N, et certains adverbes ou adjectifs (ex. *bien=pensant*, *dos=à=dos*).

Les composés sur préposition, numéral, point cardinal ou par phrase sont déclarés avec trait d'union et sans alternative, à quelques exceptions près (p.ex. on admet *je ne sais quoi* et *qu'en dira-t-on*).

On sait qu'il y a des tendances fortes dans l'usage du trait d'union ; ainsi par exemple, il est la norme pour les composés V-N. Cette norme est cependant loin d'être appliquée systématiquement (cf. p.ex. vide-greniers.org), et elle est bien souvent inutile. Qu'on écrive *garde-champêtre* ou *garde champêtre*, ces deux séquences peuvent en confiance être analysées comme des réalisations d'un même composé. Il y a des cas où, certes, la présence du trait d'union est censée signer l'emploi d'un composé par opposition à un sens littéral. C'est le cas par exemple de l'adverbe *sur-le-champ* ou de noms comme *queue-de-pie*. Pour ces cas, le DM-C contient deux entrées : la version avec trait d'union qui sera analysée de façon déterministe comme un composé (via un trait spécial de NooJ : `+UNAMB`) et une version avec espaces qui sera analysée de façon non déterministe, à la fois comme une occurrence possible du composé ou comme une séquence de chacun des mots composant la suite. Nous faisons l'hypothèse (dont la validation demanderait un travail spécifique) que

l'emploi « fautif », avec espaces, des expressions à traits d'union est en règle générale plus fréquent que l'emploi littéral des expressions correspondantes. Signalons cependant que si, pour une meilleure analyse des textes réels, le DM-C tolère la variation au niveau des formes, il propose une graphie précise au niveau des lemmes.

5 Conclusion

Le DM-C et la grammaire NooJ présentés ici sont téléchargeables sur le site du LRL et utilisables dans les termes de la licence GPL : http://lrl.univ-bpclermont.fr/spip.php?rubrique48. Avec ces deux ressources, on pense améliorer la couverture lexicale des composés, et apporter des éléments de solution au problème de la variabilité des usages de flexion et du trait d'union, ainsi qu'à celui de la productivité des schémas de composition. Il s'agit d'*éléments* de solution, mais circonscrire un espace où une analyse fiable peut être menée est en soi un résultat. 4035 formes des dictionnaires DELA, Lefff, Lexique et Multext réunis sont inconnues du DM-C. La grammaire en analyse 2682 (66 %). L'examen de ces analyses ne révèle que 7 erreurs : l'abréviation *c-à-d* comme N-GP, *avant-gauche* comme PREP-N et cinq composés sur des adjectifs de couleurs vus comme des composés A-N (ex. *blanc-bleu*). L'évaluation sur corpus des deux ressources serait un travail de recherche en soi. Elle nécessiterait de prendre en compte des variations selon les types de textes et les époques, susceptibles de mettre en cause la présence de certains composés dans le dictionnaire. Nous espérons que nos deux ressources pourront servir à mieux cerner ce qui, dans la composition, relève de la grammaire ou de la lexicalisation.

Références

CATACH N. (1981). *Orthographe et lexicographie. Les mots composés.* Paris : Nathan.

CATACH N. (1991). *L'orthographe en débat.* Paris : Nathan.

COURTOIS B., SILBERZTEIN M. (1990). *Dictionnaires électronique du français. Langue française* 87. Paris : Larousse.

DUBOIS J., DUBOIS-CHARLIER F. (2001). Composition et préfixation en français. Aix-en-Provence : chez les auteurs.

MATHIEU-COLAS M. (1995a). Un dictionnaire électronique des mots à trait d'union. *Langue française* 108, 76-85. Paris : Larousse.

MATHIEU-COLAS M. (1995b). « Syntaxe du trait d'union : Structures complexes ». *Linguisticæ Investigationes* XIX:1, 153-171. Amsterdam : John Benjamins B.V.

MATHIEU-COLAS M. (1996). « Essai de typologie des noms composés français ». *Cahiers de lexicologie* 69, 71-125.

NEW B. (2006). « Lexique 3 : Une nouvelle base de données lexicales. » *Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2006)*, Louvain, Belgique. http://www.lexique.org/ (version 3.80)

ROMARY L., SALMON-ALT S., FRANCOPOULO G. (2004). "Standards going concrete: from LMF to Morphalou". *Workshop on Electronic Dictionaries*, Coling 2004, Geneva.
www.cnrtl.fr/lexiques/morphalou/ (ATILF/Nancy Université - CNRS)

SAGOT B. (2010). "The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French". *7th International Conference on Language Resources and Evaluation (LREC 2010)*, Malte.
http://alpage.inria.fr/~sagot/lefff.html (version extensionnelle 3.0)

SAJOUS F., HATHOUT N., CALDERONE B. (2013). « GLÀFF, un Gros Lexique À tout Faire du Français ». *Actes de la conférence Traitement Automatique des Langues Naturelles (TALN 2013).*

SILBERZTEIN M. (1990). « Le dictionnaire électronique des mots composés. » *Langue française* 87, 71-83. Paris : Larousse. http://infolingu.univ-mlv.fr/ > *Données linguistiques* > *Dictionnaire* > *Téléchargement*

SILBERZTEIN M. (2003). NooJ Manual. http://www.nooj4nlp.net.

VÉRONIS J. (1998). *Multext-Lexicons. A set of Electronic Lexicons for European Languages*. ELRA Catalogue (http://catalog.elra.info), MULTEXT Lexicons, ref.: ELRA-L0010.