TALN-RÉCITAL 2013

TALN : Traitement Automatique des Langues Naturelles

RÉCITAL : Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues

---

Actes de la conférence TALN-RÉCITAL 2013

Volume 2 : RÉCITAL 2013

Éditeurs

Florian Boudin

Loïc Barrault

17 au 21 juin 2013

Les Sables d'Olonne, France

Sous l'égide de l'ATALA (Association pour le Traitement Automatique des langues).

Avant-propos

Il est maintenant une tradition dans la communauté de l'ATALA de venir fouler tous les dix ans les côtes à l'ouest de la France. Ainsi après la Côte d'Amour en 2003, nous sommes heureux d'accueillir sur la Côte de Lumière la 20^e conférence TALN et la 15^e édition de RÉCITAL.

L'organisation de TALN et RÉCITAL 2013 a été assurée par les équipes TALN du LINA (Laboratoire d'Informatique de Nantes Atlantique) et LST du LIUM (Laboratoire d'Informatique de l'Université du Maine). Cette organisation conjointe est une bonne illustration de la synergie de ces deux équipes mais aussi de la dynamique du TALN dans la région des Pays de la Loire.

Cette année, avec 127 soumissions à TALN (dont 70 articles longs et 57 articles courts), la conférence a confirmé une fois encore son attractivité. Le processus d'évaluation, qui a demandé un travail important, a été réalisé consciencieusement pour arriver à une sélection de 36 articles longs et 35 articles courts. Nous remercions chaleureusement les membres des comités de lecture et de programme de TALN pour le travail réalisé. Outre ces communications, 13 démonstrations et 4 ateliers viennent accompagner la conférence. La conférence sera aussi ponctuée par l'intervention de deux conférenciers invités : Josiane Mothe et Alexander Fraser que nous tenons aussi à remercier de leur présence.

Poursuivant sur la démarche initiée lors de l'édition précédente, la conférence RÉCITAL a ciblé un large spectre de publications (états de l'art, travaux préliminaires, etc.). L'accent a une nouvelle fois été mis sur la pédagogie et l'échange direct en fournissant des relectures explicatives et non anonymes aux auteurs. Cette formule fonctionne bien puisque l'on recense un total de 25 soumissions parmi lesquelles 18 ont été sélectionnées (6 présentations orales et 12 posters). Nous remercions les membres du comité de programme de RÉCITAL pour les précieux retours qui, nous en sommes convaincus, sont très appréciés des jeunes chercheurs en TALN.

Cette conférence sur un site distant, mais presque à mi-chemin entre Le Mans et Nantes, a nécessité un travail d'organisation important. Que les membres du comité d'organisation trouvent ici la reconnaissance du travail réalisé.

Nous n'oublions pas non plus les partenaires institutionnels et privés qui se sont joints à nous pour faire de cette 20^e conférence TALN et de cette 15^e édition de RÉCITAL une véritable réussite.

Comme il est de tradition aux Sables d'Olonne, nous souhaitons à l'ensemble des conférenciers « bon vent ».

Emmanuel Morin
Yannick Estève
Organisateurs de TALN 2013

Florian Boudin
Loïc Barrault
Organisateurs de RÉCITAL 2013

Comité d'organisation de TALN-RÉCITAL

Président de TALN

Emmanuel Morin LINA Université de Nantes, France

Vice-Président de TALN

Yannick Estève LIUM Université du Maine, France

Présidents de RÉCITAL

Florian Boudin LINA Université de Nantes
Loïc Barrault LIUM Université du Maine

Membres

Denis Béchet	LINA	Université de Nantes
Fethi Bougares	LIUM	Université du Maine
Adrien Bougouin	LINA	Université de Nantes
Nathalie Camelin	LIUM	Université du Maine
Béatrice Daille	LINA	Université de Nantes
Colin De La Higuera	LINA	Université de Nantes
Paul Deléglise	LIUM	Université du Maine
Estelle Delpech	LINA	Université de Nantes
Alexandre Dikovsky	LINA	Université de Nantes
Chantal Enguehard	LINA	Université de Nantes
Rima Harastani	LINA	Université de Nantes
Mohamed Hatmi	LINA	Université de Nantes
Amir Hazem	LINA	Université de Nantes
Nicolas Hernandez	LINA	Université de Nantes
Firas Hmida	LINA	Université de Nantes
Christine Jacquin	LINA	Université de Nantes
Ophelie Lacroix	LINA	Université de Nantes
Antoine Laurent	LIUM	Université du Maine
Elizaveta Loginova	LINA	Université de Nantes
Daniel Luzzati	LIUM	Université du Maine
Sylvain Meignier	LIUM	Université du Maine
Laura Monceaux	LINA	Université de Nantes
Simon Petitrenaud	LIUM	Université du Maine
Emmanuel Planas	LINA	Université Catholique de l'Ouest
Solen Quiniou	LINA	Université de Nantes
Anne-Françoise Quin	LINA	Université de Nantes
Holger Schwenk	LIUM	Université du Maine
James Scicluna	LINA	Université de Nantes
Christophe Servan	LIUM	Université du Maine
Prajol Shrestha	LINA	Université de Nantes
Déborah Sourdillat	LINA	Université de Nantes
Annie Tartier	LINA	Université de Nantes

Comité de programme RÉCITAL

Florian Boudin	LINA, Université de Nantes
Loïc Barrault	LIUM, Université du Maine
Adrien Lardilleux	Affinity Engine
Amir Hazem	LINA, Université de Nantes
Anne Vilnat	LIMSI, Université Paris-Sud
Antoine Jean-Yves	LI, Université de Tours
Aurélien Max	LIMSI, Université Paris-Sud
Benoit Favre	LIF, Université d'Aix-Marseille
Cécile Fabre	ERSS, Université Toulouse 2
Cédric Lopez	VISEO, Grenoble
Charlotte Roze	ALPAGE, INRIA Paris–Rocquencourt & Univ. Paris 7
Christian Raymond	IRISA, Université de Rennes
Christine Jacquin	LINA, Université de Nantes
Christophe Servan	LIUM, Université du Maine
Crabbé Benoît	ALPAGE, INRIA Paris–Rocquencourt & Univ. Paris 7
Denis Maurel	LI, Université de Tours
Denis Bechet	LINA, Université de Nantes
Didier Schwab	LIG, Universite Grenoble 2
Farah Benamara	IRIT, Université de Toulouse
Guy Lapalme	RALI, Université de Montréal, Canada
Jorge Mauricio Molina Mejia	LIDILEM, Université Stendhal – Grenoble 3
Juan-Manuel Torres-Moreno	LIA, Université d'Avignon et des Pays de Vaucluse
Mathieu Lafourcade	LIRMM, Université de Montpellier 2
Mathieu Roche	LIRMM, Université de Montpellier 2
Nicolas Hernandez	LINA, Université de Nantes
Nuria Gala	LIF, Université d'Aix-Marseille
Olivier Kraif	LIDILEM, Université Grenoble 3
Philippe Langlais	RALI, Université de Montréal, Canada
Raphael Rubino	School of Computing, Dublin City University, Irelande
Romain Deveaud	LIA, Université d'Avignon et des Pays de Vaucluse
Solen Quiniou	LINA, Université de Nantes
Stéphane Huet	LIA, Université d'Avignon et des Pays de Vaucluse
Thierry Hamon	LIM&BIO, Université Paris 13
Violaine Prince	LIRMM, Université Montpellier 2
Yayoi Nakamura-Delloye	LCAO, Université Paris 7

Partenaires

Or

Argent

Bronze

Institutionnels

Table des matières

Acquisition de lexique bilingue d'expressions polylexicales: une application à la traduction automatique statistique

Dhouha Bouamor
CEA-LIST, LVIC, F91191 Gif sur Yvette Cedex, France
LIMSI-CNRS, F-91403 Orsay, France
Univ. Paris Sud, Orsay, France
dhouha.bouamor@cea.fr

RÉSUMÉ

Cet article décrit une méthode permettant d'acquérir un lexique bilingue d'expressions polylexicales (EPLs) à partir d'un corpus parallèle français-anglais. Nous identifions dans un premier temps les EPLs dans chaque partie du corpus parallèle. Ensuite, nous proposons un algorithme d'alignement assurant la mise en correspondance bilingue d'EPLs. Pour mesurer l'apport du lexique construit, une évaluation basée sur la tâche de Traduction Automatique Statistique (TAS) est menée. Nous étudions les performances de trois stratégies dynamiques et d'une stratégie statique pour intégrer le lexique bilingue d'expressions polylexicales dans un système de TAS. Les expériences menées dans ce cadre montrent que ces unités améliorent significativement la qualité de traduction.

ABSTRACT

Mining a Bilingual Lexicon of MultiWord Expressions : A Statistical Machine Translation Evaluation Perspective

This paper describes a method aiming to construct a bilingual lexicon of MultiWord Expressions (MWEs) from a French-English parallel corpus. We first extract monolingual MWEs from each part of the parallel corpus. The second step consists in acquiring bilingual correspondences of MWEs. In order to assess the quality of the mined lexicon, a Statistical Machine Translation (SMT) task-based evaluation is conducted. We investigate the performance of three dynamic strategies and of one static strategy to integrate the mined bilingual MWEs lexicon in a SMT system. Experimental results show that such a lexicon significantly improves the quality of translation.

MOTS-CLÉS : Expression polylexicale, alignement bilingue, traduction automatique statistique.

KEYWORDS: MultiWord expression, bilingual alignment, statistical machine translation.

1 Introduction

Une expression polylexicale (EPL, en anglais *multiword expression*) peut être définie comme une combinaison de mots pour lesquels les propriétés syntaxiques ou sémantiques de l'expression entière ne peuvent pas être obtenues à partir de ses parties (Sag *et al.*, 2002). Les EPLs regroupent les expressions figées et semi-figées (ex. *cordon bleu*), les collocations (ex. *chemin de fer*), les entités nommées (ex. *New York*), les verbes à particule (ex. *grow up*), les constructions à verbe

support (ex. *faire face à*), etc.(Sag *et al.*, 2002; Constant *et al.*, 2011). Elles sont fréquemment employées dans les textes écrits étant donnée qu'elles constituent une part significative du lexique d'une langue. Jackendoff (1997) estime que la fréquence de leur utilisation est équivalente à celle des mots simples. Bien qu'elles soient facilement employées et reconnues par les humains, leur identification pose un problème majeur pour diverses applications du traitement automatique des langues.

Pour la Traduction Automatique Statistique (TAS), diverses améliorations ont été obtenues avec l'émergence des approches à base de segments (*phrase based approaches* en anglais) (Koehn *et al.*, 2003). Ces segments sont définis comme étant de simples n-grammes systématiquement traduits dans un corpus parallèle sans aucune motivation linguistique. Dans de tels systèmes, le manque d'un traitement adéquat des EPLs pourrait affecter la qualité de la traduction. En effet, la traduction littérale d'une expression non reconnue par le système de traduction comme une EPL constitue une cause principale à une traduction erronée et incompréhensible. Par exemple, un tel système proposera *« way of iron »* comme traduction pour *« chemin de fer »* au lieu de *« railway »*. Il est donc important d'utiliser un lexique dans lequel les EPLs sont prises en compte. Or un des points faibles des lexiques est souvent le manque de couverture pour ces unités (Sagot *et al.*, 2005). Ce point a été abordé dans plusieurs travaux (Fazly et Stevenson, 2007; Caseli *et al.*, 2009).

Cet article porte sur le traitement des EPLs bilingues, allant de l'acquisition automatique à partir de corpus parallèles à leur intégration dans un système de TAS. Nous considérons toute séquence contiguë non compositionnelle, appartenant à l'une des classes définies par (Luka *et al.*, 2006), comme une EPL. Ces unités ont été classées dans trois classes, sur la base de leurs propriétés catégorielles, ainsi que de leur degré de figement syntaxique et sémantique. Les classes sont constituées de *mots composés*, d'*expressions idiomatiques* et de *collocations*. Intuitivement, les EPLs bilingues sont utiles pour améliorer les résultats de la TAS. Cependant, des recherches plus approfondies sont nécessaires pour trouver la meilleure façon d'intégrer ce type d'unités dans ces systèmes. Dans cette étude, nous considérons la TAS comme un mode d'évaluation extrinsèque de l'utilité des EPLs et explorons différentes stratégies d'intégration de ces unités dans un système de TAS. Étant donné un lexique bilingue d'EPLs, nous proposons (1) trois stratégies d'intégration dynamiques où nous cherchons à modifier le modèle de traduction de différentes façons pour une prise en considération des EPLs bilingues et (2) une stratégie d'intégration statique dans laquelle nous incorporons ces unités sans changer le modèle de traduction.

Le reste de l'article est organisé comme suit : dans la section 2, nous passons en revue les principaux travaux en rapport avec la tâche d'extraction de traduction pour les EPLs. Puis, nous décrivons, dans la section 3, l'approche utilisée pour identifier ces unités et présentons, par la suite, l'algorithme d'alignement que nous avons implémenté. Dans la section 4, nous décrivons les stratégies d'intégration proposées. La section 5 est consacrée aux expériences menées ainsi qu'à la présentation des résultats obtenus. Nous concluons notre article par une présentation des principales perspectives en section 6.

2 État de l'art

Au cours des dernières années, de nombreux travaux de recherche ont été menés sur la tâche d'extraction bilingue d'EPLs à partir de corpus parallèles. La plupart d'entre eux commencent

tout d'abord par identifier les EPLs dans chaque partie du corpus parallèle, ensuite, se basent sur différentes techniques d'alignement pour les apparier. Les techniques d'extraction monolingue d'EPLs tournent autour de trois approches : (1) des méthodes symboliques reposant sur des patrons morphosyntaxiques (Okita *et al.*, 2010; Dagan et Church, 1994) ; (2) des méthodes statistiques utilisant des mesures d'association pour classer les EPLs candidates (Vintar et Fisier, 2008) et (3) des méthodes hybrides combinant (1) et (2) (Seretan et Wehrli, 2007; Daille, 2001). Aucune des approches n'est sans limitations. Il est difficile d'appliquer des méthodes symboliques à des données sans annotations morphosyntaxiques. En ce qui concerne les méthodes statistiques, bien qu'elles soient conçues pour des bigrammes, elles exigent la définition d'un seuil à partir duquel un segment extrait peut être considéré comme une EPL.

Pour identifier des correspondances entre expressions dans différentes langues, quelques travaux font appel à des outils d'alignement de mots simples pour guider l'alignement d'EPLs (Dagan et Church, 1994). D'autres se basent sur des algorithmes d'apprentissage statistique comme par exemple l'algorithme itératif de ré-estimation *Expectation Maximization* (Kupiec, 1993; Okita *et al.*, 2010). Une hypothèse largement suivie pour acquérir des EPLs bilingues est qu'une expression dans une langue source garde la même structure syntaxique que son équivalente dans une langue cible donnée (Seretan et Wehrli, 2007; Tufis et Ion, 2007). Or, les EPLs ne se traduisent pas forcément par des expressions ayant la même catégorie grammaticale (i.e *« insulaire en développement »* et *« small island developing »*) ou la même longueur[1] (i.e *« en ce qui concerne »* et *« as regards »*).

Le but principal de la majorité des travaux de recherche menés sur cet objet linguistique était l'acquisition *a priori* de correspondances entre les paires d'unités textuelles pour l'enrichissement de ressources lexicales. En comparaison, peu de travaux ont été réalisés sur l'exploitation de telles ressources, afin de rendre possible leur intégration dans des applications clés, telles que la désambiguïsation sémantique (Finlayson et Kulkarni, 2011) ou la recherche d'information interlingue (Vechtomova, 2005). En TAS, Lambert et Banchs (2005) introduisent une méthode dans laquelle les EPLs sont considérées comme un élément unique dans le corpus d'apprentissage. En exploitant un corpus de petite taille, ils ont montré que la qualité de l'alignement et la précision de traduction ont été améliorées. Cependant, ils ont obtenu, dans des études plus récentes (Lambert et Banchs, 2006), basées sur un corpus de taille importante, un score BLEU (Papineni *et al.*, 2002) plus bas. Nous citons notamment les travaux de (Ren *et al.*, 2009) qui implémentent une méthode permettant d'intégrer des termes multi-mots issus du domaine médical dans MOSES (Koehn *et al.*, 2007). Leur méthode a permis de gagner 0,17 points de score BLEU par rapport au système de référence. La présente étude est une extension de l'approche présentée dans (Bouamor *et al.*, 2011). Nous proposons tout d'abord une méthode ayant pour but d'extraire et aligner des EPLs bilingues. Nous étudions ensuite l'impact de l'utilisation de ces unités dans un système de TAS.

3 Acquisition de lexique bilingue d'EPLs

Dans cette section, nous décrivons l'approche proposée pour extraire un lexique bilingue d'EPLs à partir d'un corpus parallèle français-anglais aligné au niveau de la phrase. Cette approche est réalisée en deux étapes. Dans la première étape, nous identifions les EPLs dans chaque partie

[1] La longueur d'une EPL est calculée en nombre de mots

du corpus parallèle. La deuxième étape consiste en l'acquisition de correspondances bilingues d'EPLs.

3.1 Extraction monolingue d'EPLs

La méthode d'identification monolingue d'EPLs est fondée sur une approche symbolique, très similaire à celle présenté dans (Okita *et al.*, 2010). Là où ils définissent des patrons pour extraire seulement des syntagmes nominaux, notre approche identifie à la fois des syntagmes nominaux, des expressions figées et des entités nommées. La méthode proposée requiert simplement une analyse morphosyntaxique des textes source et cible, comme étape préliminaire à la procédure de construction d'expressions. Nous faisons donc appel à la plate forme d'analyse multilingue LIMA du CEA-LIST (Besançon *et al.*, 2010), qui produit une liste de lemmes étiquetés par leurs catégories grammaticales. Le processus d'identification d'EPLs opère sur des lemmes plutôt que sur des formes de surface.

Comme la plupart des expressions sont constituées de combinaisons de noms,d'adjectifs ou encore de prépositions, nous produisons une liste de n-grammes candidats ($2 \leq n \leq 4$), dont la structure morphosyntaxique respecte une configuration prédéfinie, telle que celles décrites dans le tableau 1. seize configurations ont été manuellement définies. Notons qu'il existe des patrons d'extraction (ou configurations) pour lesquels aucune EPL n'a été produite (c-à-d. Past_Participle-Noun). Un tel type d'analyse permet de ne garder que des n-grammes jugés pertinents et d'écarter ceux constitués de mots vides comme par exemple *« is a, of the, de la »*.

Configuration	**Exemples anglais/français**
Adj-Noun	Plenary meeting/Libre circulation
Noun-Adj	Oil tanker/Parlement européen
Noun-Noun	Member state/Etat membre
Past_Participle-Noun	Developped country/...
Noun-Past_Participle	Parliament adopted/Pays developpé
Adj-Adj-Noun	European public prosecutor/...
Adj-Noun-Adj	Social market economy/Bon conduite administratif
Adj-Noun-Noun	Renewable energy source/...
Noun-Noun-Adj	.../Industrie automobile allemand
Noun-Adj-Adj	.../Ministère public européen
Adj-Noun-Adj	Social fund assistance/Important débat politique
Noun-Prep-Noun	Point of view/Chemin de fer
Noun-Prep-Adj-Noun	Court of first instance/Court de premier instance
Noun-Prep-Noun-Adj	.../Source d'énergie renouvelable
Adj-Noun-Prep-Noun	European court of justice/...
Noun-Adj-Prep-Noun	.../Politique européen de concurrence

TABLE 1 – Configurations morphosyntaxiques permises.

A cette liste de candidats, nous ajoutons des expressions idiomatiques prépositionnelles comme par exemple *« in the light of, with regard to, en ce qui concerne... »* et des entités nommées telles que *« Middle East, South Africa, El-Salvador... »* reconnues par la plate-forme LIMA. Le résultat

ID.PHRASE	PHRASE
2	*. . .semblerait être* ***à nouveau*** *mis en accusation, le ministère public . . .*
55	*. . .vous demande donc* ***à nouveau*** *de faire le nécessaire . . .*
$n-1$	*. . .aussi de promouvoir* ***à nouveau*** *l'activité des femmes . . .*
n	*. . .que le règlement soit* ***à nouveau*** *modifié en collaboration . . .*

⇓

	1	2	3	4	.....	55	.....	$n-1$	n
à nouveau	0	1	0	0	.....	1	.....	1	1

FIGURE 1 – Représentation vectorielle de l'expression *« à nouveau »*. ID.PHRASE correspond à un identifiant unique de la phrase contenant l'expression dans notre corpus.

de l'extraction est représenté par une liste de candidats triée par ordre décroissant selon leur fréquence dans le corpus. Plusieurs candidats parmi ceux produits apparaissent dans d'autres candidats. Afin d'éviter un effet de surgénération, nous proposons les heuristiques de nettoyage suivantes :

- Si une expression est imbriquée dans une autre et qu'elles ont la même fréquence, on ne garde que la plus couvrante (plus longue).
- Si une expression apparaît dans un grand nombre d'autres expressions, nous suivons l'approche proposée par (Frantzi *et al.*, 2000) et éliminons toutes les expressions plus longues.

À la différence de beaucoup d'autres systèmes existants (Daille, 2001; Seretan et Wehrli, 2007; Vintar et Fisier, 2008), notre système n'applique pas de filtre fondé sur des mesures d'association ou sur la fréquence. Nous prenons en considération toutes les expressions extraites, aussi bien fréquentes que non fréquentes et celles dont le degré de corrélation entre ses constituants est élevé ou faible. A notre connaissance, aucune approche n'a pris en considération tout l'ensemble.

3.2 Méthode d'alignement

Dans cette section, nous présentons une méthode qui tente de trouver, pour chaque expression source, la traduction qui lui est adéquate dans l'ensemble d'expressions cibles. Cette tâche pose de sérieux problèmes en l'absence de ressources externes comme les dictionnaires bilingues ou les outils d'alignement de mots simples. Nous proposons une méthode indépendante de toute ressource externe, qui requiert simplement un corpus parallèle et la liste de candidats source et cible à traduire. Notre approche hérite de la sémantique distributionnelle, où nous associons à chaque expression source et cible une représentation spécifique qui servira par la suite de base pour l'établissement d'une relation de traduction entre chaque paire d'expressions (source, cible). Nous faisons appel au modèle vectoriel (Salton *et al.*, 1975), un modèle algébrique souvent utilisé en recherche d'information. Nous représentons chaque expression par un vecteur de dimension n (nombre de phrases dans le corpus) indiquant si elle apparaît ou non dans chaque phrase du corpus. La figure 1 décrit le vecteur représentant l'EPL française *« à nouveau »*.

Pour extraire des paires de traduction d'EPLs, nous proposons un algorithme itératif d'alignement opérant de la façon suivante :

1. Trouver l'expression la plus fréquente dans chaque phrase source.

Français	→	Anglais
parlement européen	→	european parliament
état par état	→	amount of state
coup d'état	→	military coup
zone non fumeur	→	no smoking area
insulaire en développement	→	small island developing
de bonne foi	→	good faith
politique de concurrence	→	competition policy
chemin de fer	→	railway sector
en ce qui concerne	→	in regard to
en ce qui concerne	→	as regards
en ce qui concerne	→	with reference to
en ce qui concerne	→	with respect to
coupe forestier	→	cut in forestation

TABLE 2 – Exemples d'EPLs bilingues alignées par l'algorithme décrit ci-dessus.

2. Extraire les expressions cibles qui apparaissent dans toutes les phrases parallèles à celles où figure l'expression source.
3. Calculer un score de confiance pour chaque couple (source, cible).
4. Considérer l'expression cible qui maximise ce score comme la meilleure traduction.
5. Supprimer la paire de traductions du processus et retourner vers 1.

Le score de confiance est calculé sur la base de la mesure de l'indice de Jaccard (équation 1).

$$\text{Jaccard} = \frac{I_{st}}{V_s + V_t - I_{st}} \tag{1}$$

Cette mesure est fondée principalement sur le calcul de nombre de phrases partagées par chaque expression source et cible nommé ici I_{st} qu'on normalise par la somme des normes des vecteurs V_s et V_t diminué de l'ensemble d'intersection.

En observant certaines paires d'expressions du tableau 2, nous remarquons que notre méthode présente plusieurs avantages. Premièrement, pour trouver la traduction adéquate pour chaque EPL et contrairement à la plupart des travaux antérieurs (Dagan et Church, 1994; Ren *et al.*, 2009) qui reposent sur la traduction mot à mot des composantes d'une EPL, notre méthode capture l'équivalence sémantique entre les EPLs en n'ayant recours à aucune information préalable sur l'alignement des mots. Elle permet aussi d'aligner des expressions à caractère idiomatique tel que *« à nouveau → once more »* ou encore *« état par état → amount of state »* et trouve toutes les correspondances bilingues possibles pour les EPLs source pour lesquelles plusieurs EPLs cible correctes existent. Par exemple, notre méthode fournit pour l'EPL *« en ce qui concerne »* les traductions suivantes : *« in regard to », « with reference to », « with respect to », « as regards »*.

Nous avons pu aussi identifier une classe d'erreurs dont la cause provient essentiellement du choix de la taille des n-grammes. Comme nous ne prenons en considération que des alignements *m-n* avec m≥2 and n≥2, quelques expressions dont la traduction de référence est constituée

d'un seul mot ne sont pas alignées correctement. Par exemple, l'expression française *« chemin de fer »* correspondant normalement au mot simple anglais *« railway »* est traduite par l'expression *« railway sector »*.

4 EPLs dans Moses

Dans la section précédente, nous avons décrit l'approche suivie pour acquérir un lexique bilingue d'EPLs. Pour évaluer sa qualité, nous avons mené dans (Bouamor *et al.*, 2011) une évaluation intrinsèque à petite échelle dans laquelle nous comparons les paires d'EPLs bilingues acquises à un alignement de référence créé manuellement. Sur un corpus de test constitué de 100 paires de phrases parallèles issues du corpus Europarl, nous avons obtenu une précision de 63,93%, un rappel de 62,46% et une F-mesure de 63,19%. Comme il n'existe à ce jour aucun protocole commun permettant d'évaluer les résultats d'alignement d'EPLs, nous conduisons une évaluation extrinsèque dans laquelle nous étudions l'impact de l'utilisation de ces unités dans Moses, un système de TAS à base de segments. Néanmoins, comme mentionné dans la section 1, la difficulté consiste à trouver la meilleure façon d'intégrer les EPLs dans de tels systèmes. À cet effet, nous proposons trois *stratégies d'intégration dynamiques* où le modèle de traduction est modifié de différentes façons et une *stratégie d'intégration statique* dans laquelle nous introduisons les EPLs au décodeur sans changer le modèle de traduction et comparons leurs performances dans la section 5.

4.1 Moses comme système de référence

Le système de traduction de référence (**Réf**) utilisé est Moses, un outil sous licence libre. Dans ce système, l'unité de traduction est le segment, qui correspond à un groupe de mots contigus. Le modèle de traduction sert de pont entre les langues source et cible. Son rôle est de guider la construction, pour chaque phrase source, d'un ensemble d'hypothèses de traduction en langue cible. Lors de la phase de décodage, ces hypothèses de traduction sont sélectionnées à partir d'un inventaire constitué d'un ensemble d'appariements entre des segments de longueur variable. Ces associations et les scores qui les accompagnent constituent la table de traduction *(phrase table)*.

4.2 Stratégies d'intégration dynamiques

4.2.1 Nouveau modèle de traduction

Les tables de traduction constituent la source principale de connaissance pour le décodeur. Le décodeur consulte ces tables pour déterminer comment traduire une phrase source en langue cible. Cependant, en raison d'erreurs d'alignement automatique de certains mots, des segments extraits peuvent être dénués de sens. Pour remédier à ce problème, nous proposons de considérer les EPLs comme des paires de phrases parallèles : nous les ajoutons au corpus d'apprentissage et entraînons un nouveau modèle de traduction. Dans cette méthode (**Train**), nous espérons que par l'augmentation du nombre d'occurrences des paires d'EPLs, considérées comme de bons segments, une modification de l'alignement et de la probabilité de la traduction soit enregistrée.

4.2.2 Extension de la table de traduction

Dans cette méthode, nous étendons la table de traduction du système de référence RÉF en y incorporant les paires d'EPLs bilingues acquises. Nous utilisons la valeur de l'indice de Jaccard proposée pour chaque paire d'EPLs pour définir la probabilité de traduction dans les deux directions et fixons les probabilités lexicales à 1 pour des raisons de simplicité. Ainsi, le décodeur prendra en considération des EPLs bilingues lors de la recherche de segments candidats pour traduire une phrase source. Cette méthode est notée **TABLE** dans le reste de cet article.

4.2.3 Trait additionnel pour les EPLs

(Lopez et Resnik, 2006) ont souligné qu'une meilleure définition des traits utilisés peut conduire à un gain substantiel dans la qualité des traductions. Nous suivons cette hypothèse et étendons la méthode TABLE. Nous définissons un nouveau *trait binaire* indiquant pour chaque entrée de la table de traduction s'il s'agit d'une EPL ou pas. Le but de cette méthode notée **TRAIT** est de guider le système pour choisir les EPLs bilingues plutôt que les hypothèses proposées par RÉF.

4.3 Stratégie d'intégration statique

Dans cette méthode, notée **FORCÉ**, nous voulons que le décodeur prenne en considération des EPLs bilingues tout en gardant le modèle de traduction de RÉF. À cet égard, nous utilisons le *mode de décodage forcé* du décodeur de MOSES. Ce dernier comporte un schéma de balisage XML permettant de spécifier des traductions pour des parties des phrases à traduire. Nous pouvons ainsi indiquer au décodeur ce qu'il faut utiliser pour traduire certains mots ou segments dans les phrases à traduire. Dans le cadre de notre étude, nous représentons chaque EPL apparaissant dans le corpus de test par la balise XML adéquate en se basant sur les traductions produites par notre aligneur. Un exemple de représentation de l'EPL *« à nouveau »* est présenté ci-dessous :

. . .sembler être **à nouveau** mis en accusation, le ministère public . . .

⇓

. . .sembler être < ***mwe*** **translation=*"once more***" >**à nouveau**< /***mwe***> mis en accusation, le ministère public . . .

5 Expériences et résultats

5.1 Corpus et outils

Les données d'apprentissage et de test proviennent du corpus Europarl pour la paire de langues français-anglais. Ce corpus regroupe un ensemble de phrases parallèles extraites des actes du parlement européen. Pour estimer le modèle de traduction du système de référence RÉF, nous avons construit un corpus d'apprentissage contenant après normalisation 100 000 paires de phrases. La normalisation est établie à travers les traitements suivants : tokenisation, suppression de phrases de plus de 50 mots et lemmatisation à l'aide de l'outil TreeTagger. Nous utilisons

Method	BLEU		TER	
	Tous_Test	EPLs_*Test*	*Tous_Test*	EPLs_*Test*
RÉF	28,85	30,83	55,44	53,59
Dynamiques				
TRAIN	**28,87**	**31,06**	**55,38**	**53,32**
TABLE	28,82	**30,88**	**55,42**	**53,46**
TRAIT	**28,95**	**31,06**	55,48	**53,56**
Statique				
FORCÉ	28,20	29,19	56,01	55,05

TABLE 3 – Résultats de traduction des corpus de test *Tous_Test* et EPLs_*Test* en termes de scores BLEU et TER

ce même corpus pour construire un lexique bilingue d'EPLs[2]. Comme les entrées de ce lexique sont sous forme de lemmes et que le mode de décodage forcé de MOSES n'est actuellement pas compatible avec les modèles à base de facteurs, le modèle de traduction a été estimé sur des lemmes plutôt que sur des formes de surface. Outre le modèle de traduction, nous avons entraîné un modèle de langue (trigramme) sur une version lemmatisée de la totalité du corpus Europarl (1,8M) en utilisant la boite à outils IRSTLM[3].

Les stratégies dynamiques et statique décrites précédemment sont ensuite appliquées. Dans TRAIN, les EPLs bilingues sont ajoutées au corpus d'apprentissage pour estimer un nouveau modèle de traduction. En ce qui concerne TABLE, la table de traduction de RÉF est enrichie par les EPLs bilingues. Dans TRAIT, un trait additionnel 1/0 est introduit dans la table de traduction de TABLE. Finalement, FORCÉ maintient le modèle de traduction de RÉF. Tous les modèles obtenus sont optimisés par minimisation du taux d'erreur (*MERT : Minimum Error Rate Training*) (Och, 2003) sur un corpus de développement constitué de 4 000 paires de phrases issues du même corpus.

5.2 Résultats et discussion

Deux séries d'expériences ont été menées : *Tous_Test* et EPLs_*Test*. Le premier corpus de test *Tous_Test* est constitué de 1 000 paires de phrases parallèles extraites aléatoirement du corpus Europarl. Pour mesurer l'apport réel du lexique bilingue d'EPLs, nous avons constitué un corpus de test noté EPLs_*Test* où nous ne conservons que les phrases du corpus *Tous_Test* contenant au moins une EPL. Ce corpus contient 323 paires de phrases parallèles. La qualité de traduction du système RÉF et des différentes stratégies dynamiques et statique d'intégration est évaluée sur les deux corpus de test sur la base des mesures BLEU et TER (Snover *et al.*, 2006). Nous considérons qu'à chaque phrase source correspond une seule phrase de référence en langue cible. Les résultats de traduction pour les différentes configurations sont rassemblés dans le tableau 3.

À première vue, nous remarquons que le score BLEU varie en fonction du type du jeu de test. Concernant le corpus de test *Tous_test*, la meilleure amélioration est obtenue par la stratégie dynamique TRAIT. Cette méthode rapporte un faible gain de +0,1 points BLEU par rapport au

[2]EPLs bilingues extraites par l'algorithme décrit dans la section 3
[3]`http://hlt.fbk.eu/en/irstlm`

SRC	*je entendre en effet lancer un initiative communautaire pour le afrique en étendre le ligne nepad ...*
RÉFERENCE	*indeed , i intend to launch a community initiative* **for africa** *, develop the nepad line...*
RÉF	*i hear be indeed launch an* initiative **for the eu africa** *by extend the nepad line ...*
TRAIT	*i hear in fact launch a community initiative* **for africa** *by extend the nepad line ...*
SRC	*le deuxième groupe de problème relever de le aide international et du prochain engagement de johannesburg.*
RÉFERENCE	*another series of problem* **mention be a matter of** international aid *and the forthcoming johannesburg summit.*
RÉF	*the second group of the problem* **be a matter of** *international aid and the forthcoming johannesburg commitment.*
FORCÉ	*the second group of the problem* **relate to the** *international aid and the forthcoming johannesburg commitment.*

TABLE 4 – Exemples de traduction. Notons que le texte est lemmatisé. Nous soulignons les EPLs et mettons en gras différentes suggestions pour le contexte immédiat gauche ou droite.

système RÉF. Le premier exemple de traduction présenté dans le tableau 4 souligne la contribution du trait introduit à l'amélioration de la qualité de traduction. Contrairement à RÉF, traduisant l'EPL *« initiative communautaire »* par simplement le mot simple *« initiative »*, la stratégie TRAIT mène à bien la traduction de l'EPL *« initiative communautaire »* par *« community initiative »* et de son contexte immédiat (*« for africa »*). Des scores BLEU plus faibles que ceux rapportés par RÉF sont obtenus par les stratégies TABLE et FORCÉ.

Pour le corpus de test EPLs_*Test*, qui ne considère que les phrases contenant des EPLs du lexique bilingue, nous constatons que toutes les stratégies d'intégration dynamiques rapportent des scores BLEU plus élevés que ceux obtenus par RÉF et la stratégie statique FORCÉ. Un gain de $+0,23$ points BLEU est obtenu par TRAIT et TRAIN. La stratégie TABLE rapporte un score légèrement amélioré montrant un gain de $+0,05$ points BLEU. Contrairement aux stratégies d'intégration dynamiques, la méthode FORCÉ obtient de faibles scores sur les deux corpus de test. Ceci peut être expliqué de la manière suivante : nous supposons qu'en forçant le décodeur à traduire une EPL par une unité donnée, ce dernier échoue à bien traduire le contexte immédiat gauche ou droit de l'EPL induisant ainsi une diminution de la valeur du score BLEU. Ainsi, dans le second exemple du tableau 4, les deux systèmes produisent une bonne traduction pour l'EPL *« aide international »*. Cependant, FORCÉ échoue dans la traduction du segment *« relever de »*. Il est important de noter que cette traduction pourrait être soutenue dans le cas où nous associons à chaque phrase source de multiples traductions de référence.

Dans une étude antérieure, (Ren *et al.*, 2009) ont proposé une stratégie similaire à la stratégie TRAIT dans laquelle ils indiquent pour chaque entrée de la table de traduction si un segment contient une paire d'EPL bilingue spécialisée. Pour le domaine médical, leur méthode rapporte un gain de $+0,17$ points BLEU par rapport à MOSES, un gain plus faible que celui obtenu par la stratégie TRAIT. La question que l'on peut se poser en observant les différents résultats obtenus est : est-il possible de prétendre que le système ayant les meilleurs scores est vraiment

le meilleur système ? En d'autres termes, les résultats obtenus par différentes stratégies sont-ils *statistiquement significatifs* ?

Pour évaluer la significativité statistique des résultats obtenus, nous utilisons la *méthode par ré-échantillonage par amorce* décrite par (Koehn, 2004). Cette méthode estime la probabilité (*p-valeur*) qu'une différence mesurée entre les scores BLEU surgisse par hasard, par la création à plusieurs reprises (10 fois) d'échantillons uniformes avec remise à partir des corpus de tests. Nous nous appuyons sur cette méthode pour comparer les méthodes TRAIN, TABLE et TRAIT apportant des gains dans le score BLEU (Tableau 3) par rapport à RÉF. Les résultats obtenus sont présentés dans le tableau ci-dessous.

Méthode	*p-valeur* (95 % *IC*)	
	Tous_Test	EPLs_*Test*
RÉF	-	-
TRAIN	0,1	0,05
TABLE	-	0,3
TRAIT	0,01	0,01

TABLE 5 – Test de significativité statistique des résultats en termes de *p-valeur*

Sur un intervalle de confiance (IC) de 95%, les résultats varient de non significatifs (quand $p > 0,05$) à hautement significatifs. Sur les deux corpus de test, nous remarquons que les améliorations apportées par la stratégie TRAIT ayant une p-valeur de 0,05 sont significatifs. Cependant, le faible gain en score BLEU obtenu par TABLE (0,3 de p-valeur) est non significatif. La cause est que nous utilisons la valeur de l'indice de Jaccard, une mesure utilisée pour comparer la similarité et la diversité entre des échantillons, pour définir la probabilité de traduction. Ceci peut être ajusté par la transformation des valeurs de Jaccard obtenus pour chaque paire d'EPL en une probabilité de traduction assurant ainsi l'uniformité et la cohérence des probabilités dans la table de traduction.

Le score BLEU relève seulement les améliorations globales et ne montre aucune différence pouvant être révélée par une évaluation humaine. Cette observation nous a motivé à mener une évaluation lexicale fine des EPLs du corpus EPLs_*Test*. Nous avons construit un corpus de test constitué uniquement d'EPLs et avons manuellement créé une liste de références à partir du corpus de référence. Ce corpus a été traduit par RÉF,TRAIN, TABLE, TRAIT et FORCÉ. Les résultats obtenus évalués par les mesures du BLEU et TER sont présentés dans la figure 2. Nous constatons qu'un gain de $+9,8$ et de $-0,2$ respectivement de points BLEU et TER est relevé par la stratégie FORCÉ. Cela vient confirmer que l'obtention d'un faible score BLEU avec la stratégie FORCÉ dans les expériences précédentes n'est pas due à une mauvaise qualité du lexique bilingue d'EPLs. Nous remarquons aussi que les stratégies TRAIN et TRAIT obtiennent des scores plus élevés (respectivement 24,67 et 28,06 points BLEU) par rapport à ceux obtenus par RÉF ayant un score BLEU de 21,84.

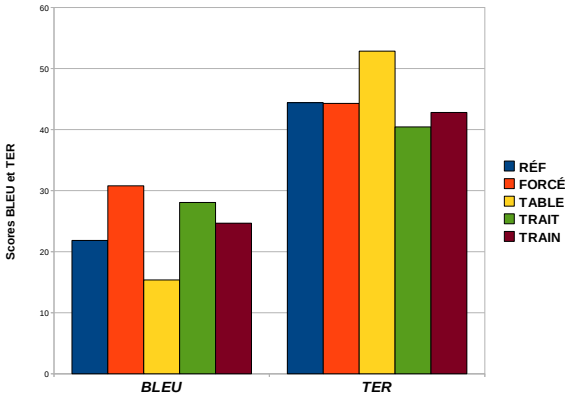


FIGURE 2 – Évaluation lexicale des EPLs en terme de scores BLEU et TER

6 Conclusion et travaux futurs

Dans cet article, nous avons décrit une méthode permettant d'extraire et d'aligner des EPLs dans un corpus parallèle français-anglais. L'algorithme d'alignement proposé effectue des alignements de type *m-n* et prend en considération des EPLs quel que soit le degré de corrélation entre leurs constituants. Pour mesurer l'apport de ces unités pour MOSES, nous avons présenté trois stratégies d'intégration dynamiques où nous avons modifié le modèle de traduction de différentes façons pour une prise en considération des EPLs bilingues et une stratégie d'intégration statique dans laquelle nous avons incorporé ces unités sans changer le modèle de traduction. Les expériences menées dans ce cadre montrent que la stratégie dynamique TRAIT, où un trait additionnel indiquant pour chaque entrée de la table de traduction s'il s'agit d'une EPL ou pas, peut améliorer significativement les résultats obtenus par MOSES avec un gain allant jusqu'à $+0,23$ points BLEU.

Nous considérons que nos expériences initiales sont positives et peuvent être améliorées de diverses façons. Dans ce présent travail, le modèle de traduction est estimé sur des lemmes plutôt que sur des formes de surface. Nous avons d'abord l'intention d'utiliser un modèle de génération pour produire les formes de surfaces adéquates à partir des résultats de traduction, présentés ici en lemmes. Nous comptons aussi entrainer notre système de traduction sur un corpus de taille plus importante afin d'évaluer l'impact du volume des données sur les résultats obtenus. En plus de leur application dans un système de TAS, nous tenterons d'étudier l'impact de ces EPLs sur la pertinence des résultats du moteur de recherche interlingue du CEA LIST.

Références

BESANÇON, R., DE CHALENDAR, G., FERRET, O., GARA, F., LAIB, M., MESNARD, O. et SEMMAR, N. (2010). Lima : A multilingual framework for linguistic analysis and linguistic resources development and evaluation. *In Proceedings of LREC*, Malta.

BOUAMOR, D., SEMMAR, N. et ZWEIGENBAUM, P. (2011). Improved statistical machine translation using multi-word expressions. *In Proceedings of MT-LIHMT*, Barcelona, Spain.

CASELI, H., VILLAVICENCIO, A., MACHADO, A. et FINATTO, M. J. (2009). Statistically-driven alignment-based multiword expression identification for technical domains. *In Proceedings of the Workshop on Multiword Expressions : Identification, Interpretation, Disambiguation and Applications*, Singapore.

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

DAGAN, I. et CHURCH, K. (1994). Termight : Identifying and translating technical terminology. *In Proceedings of the 4th Conference on ANLP*, pages 34–40, Stuttgart, Germany.

DAILLE, B. (2001). Extraction de collocation à partir de textes. *In* MAUREL, D., éditeur : *Actes de TALN 2001 (Traitement automatique des langues naturelles)*, Tours. ATALA, Université de Tours.

FAZLY, A. et STEVENSON, S. (2007). Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures. *In Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 9–16.

FINLAYSON, M. et KULKARNI, N. (2011). Detecting multi-word expressions improves word sense disambiguation. *In Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World*, pages 20–24, Portland, Oregon, USA.

FRANTZI, C., ANANIADOU, S. et MIMA, H. (2000). Automatic recognition of multi-word terms : the c-value/nc-value method. *In Int. J. on Digital Libraries 3(2)*, pages 115–130.

JACKENDOFF, R. (1997). The architecture of the language faculty. *MIT Press*.

KOEHN, P. (2004). Statistical significance tests for machine translation evaluation. *In Proceedings of EMNLP*.

KOEHN, P., HOANG, H., BIRCH, A., CALLISON-BURCH, C., FEDERICO, M., BERTOLDI, N., COWAN, B., SHEN, W., MORAN, C., ZENS, R., DYER, C., BOJAR, O., CONSTANTIN, A. et HERBST, E. (2007). Moses : Open source toolkit for statistical machine translation. *In Proceedings of ACL, demo session*, Prague, Czech Republic.

KOEHN, P., OCH, F. et MARCU, D. (2003). Statistical phrase-based translation. *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 115–124, Edmonton, Canada.

KUPIEC, J. (1993). An algorithm for finding noun phrases correspondences in bilingual corpora. *In Proceedings of the 31st annual Meeting of the Association for Computational Linguistics*, pages 17–22, Columbus, Ohio, USA.

LAMBERT, P. et BANCHS, R. (2005). Data inferred multi-word expressions for statistical machine translation. *In Proceedings of MT SUMMIT*.

LAMBERT, P. et BANCHS, R. (2006). Grouping multi-word expressions according to part-of-speech in statistical machine translation. *In Proceedings of the Workshop on Multi-word Expressions in a multilingual context*.

LOPEZ, A. et RESNIK, P. (2006). Word-based alignment, phrase based translation :what's the link ? *In Proceedings of the association for machine translation in the Americas : visions for the future of machine translation*, pages 90–99.

LUKA, N., SERETAN, V. et WEHRLI, E. (2006). Le problème de collocation en tal. *In Nouveaux cahiers de linguistiques Française*, pages 95–115.

OCH, F. (2003). Minimum error rate training in statistical machine translation. *In Proceedings of ACL*.

OKITA, T., GUERRA, M., ALFREDO GRAHAM, Y. et WAY, A. (2010). Multi-word expression sensitive word alignment. *In Proceedings of the 4th International Workshop on Cross Lingual Information Access at COLING 2010*, pages 26–34, Beijing.

PAPINENI, k., ROUKOS, S., WARD, T. et ZHU, W. J. (2002). Bleu : a method for automatic evaluation of machine translation. *In 40th Annual meeting of the Association for Computational Linguistics*.

REN, Z., LU, Y., LIU, Q. et HUANG, Y. (2009). Improving statistical machine translation using domain bilingual multiword expressions. *In Proceedings of the Workshop on Multiword Expressions : Identification, Interpretation, Disambiguation and Applications*, pages 47–57.

SAG, I., BALDWIN, T., FRANCIS BOND, F., COPESTAKE, A. et FLICKINGER, D. (2002). Multiword expressions :a pain in the neck for nlp. *In CICLing 2002*, Mexico City, Mexico.

SAGOT, B., CLÉMENT, L., DE LA CLERGERIE, É., BOULLIER, P. *et al.* (2005). Vers un méta-lexique pour le français : architecture, acquisition, utilisation. *In Actes de TALN*.

SALTON, G., WONG, A. et YANG, C. (1975). A vector space model for automatic indexing. *In Communications of the ACM*, pages 61–620.

SERETAN, V. et WEHRLI, E. (2007). Collocation translation based on sentence alignment and parsing. *In* BENARMARA, F., HATOUT, N., MULLER, P. et OZDOWSKA, S., éditeurs : *Actes de TALN 2007 (Traitement automatique des langues naturelles)*, Toulouse. ATALA, IRIT.

SNOVER, M., DORR, B., SCHWARTZ, R., MICCIULLA, L. et MAKHOUL, J. (2006). A study of translation edit rate with targeted human annotation. *In Proceedings of Association for Machine Translation in the Americas*.

TUFIS, I. et ION, R. (2007). Parallel corpora, alignment technologies and further prospects in multilingual resources and technology infrastructure. *In Proceedings of the 4th International Conference on Speech and Dialogue Systems*, pages 183–195.

VECHTOMOVA, O. (2005). The role of multi-word units in interactive information retrieval. *In ECIR2005*, pages 403–420, Berlin.

VINTAR, S. et FISIER, D. (2008). Harvesting multi-word expressions from parallel corpora. *In Proceedings of LREC*, Marrakech, Morocco.

Quelques variations sur les mesures de comparabilité quantitatives et évaluations sur des corpus comparables Français-Anglais synthétiques

Guiyao Ke[1, 2]
(1) IRISA, UMR 6074
(2) Université de Bretagne Sud, 56000 Vannes
guiyao.ke@univ-ubs.fr

RÉSUMÉ

Dans la suite des travaux de (Li et Gaussier, 2010) nous abordons dans cet article l'analyse d'une famille de mesures quantitatives de comparabilité pour la construction ou l'évaluation des corpus comparables. Après avoir rappelé la définition de la mesure de comparabilité proposée par (Li et Gaussier, 2010), nous développons quelques variantes de cette mesure basées principalement sur la prise en compte des fréquences d'occurrences des entrées lexicales et du nombre de leurs traductions. Nous comparons leurs avantages et inconvénients respectifs dans le cadre d'expérimentations basées sur la dégradation progressive du corpus parallèle Europarl par remplacement de blocs selon la méthodologie suivie par (Li et Gaussier, 2010). L'impact sur ces mesures des taux de couverture des dictionnaires bilingues vis-à-vis des blocs considérés est également examiné.

ABSTRACT

Some variations on quantitative comparability measures and evaluations on synthetic French-English comparable corpora

Following the pioneering work by (Li et Gaussier, 2010) we address in this paper the analysis of a family of quantitative measures of comparability dedicated to the construction or evaluation of comparable corpora. After recalling the definition of the comparability measure proposed by (Li et Gaussier, 2010), we develop some variants of this measure based primarily on the consideration of the occurrence frequency of lexical entries and the number of their translations. We compare the respective advantages and disadvantages of these variants in the context of an experiments based on the progressive degradation of the Europarl parallel corpus, by replacing blocks according to the methodology followed by (Li et Gaussier, 2010). The impact of the coverage of bilingual dictionaries on these measures is also discussed.

MOTS-CLÉS : Corpus comparables, Mesures de comparabilité, Évaluation.

KEYWORDS: Comparable corpora, Comparability measures, Evaluation.

1 Introduction

La notion de comparabilité entre documents est assez délicate à introduire : il est communément admis de considérer que deux documents de langues différentes sont comparables lorsque ces documents traitent de sujets analogues. Par extension, la notion de corpus comparable a été introduite par (Fung et Yee, 1998), (Munteanu *et al.*, 2004) et reste assez subjective. (Déjean et Gaussier, 2002) ont proposé une définition quantitative de cette notion de comparabilité selon laquelle : *Deux corpus de deux langues $\mathscr{L}_1$ et $\mathscr{L}_2$ sont dits comparables s'il existe une sous-partie non négligeable du vocabulaire du corpus de langue $\mathscr{L}_1$, respectivement $\mathscr{L}_2$, dont la traduction se trouve dans le corpus de langue $\mathscr{L}_2$, respectivement $\mathscr{L}_1$*. (Li et Gaussier, 2010) en ont dérivé une mesure qui s'appuie sur un dictionnaire de traduction parallèle. Ces auteurs ont proposé d'évaluer cette mesure en partant de documents parallèles (c'est-à-dire de traduction directe) puis de dégrader cette traduction en observant la variation produite sur leur mesure : l'idée principale étant de vérifier la cohérence de la mesure proposée quand le nombre de traductions directes des entrées lexicales diminue. Cette mesure est principalement basée sur un comptage de présence de traductions des entrées lexicales qui dépend d'une manière non explicitée à la fois du dictionnaire de traduction et de la composition des corpus étudiés. Dans cet article nous proposons d'étudier et de comparer deux variantes autour de cette mesure de comparabilité en introduisant des informations quantitatives supplémentaires concernant le nombre d'occurrences des entrées lexicales et le nombre de traductions associées, en conjecturant que ces deux grandeurs produiront des effets positifs dans certaines situations. Ces nouvelles mesures sont présentées puis évaluées par rapport à la mesure développée par (Li et Gaussier, 2010), en prenant en considération la couverture du dictionnaire de traduction exploité.

2 Variations autour d'une mesure quantitative de comparabilité

2.1 Mesure de comparabilité de Li et Gaussier (Cmp_{LG})

Cette mesure fait intervenir un comptage du nombre des entrées lexicales *passerelles* permettant de *coupler* deux corpus de langues distinctes via un lexique de traduction. Notons C_1 un corpus en langue $\mathscr{L}_1$ et C_2 un corpus en langue $\mathscr{L}_2$. La mesure de similarité définie par (Li et Gaussier, 2010) se présente formellement sous la forme :

$$Cmp_{LG}(C_1, C_2) = \frac{\sum_{w_1 \in WC_1 \cap WD_1} \sigma(w_1) + \sum_{w_2 \in WC_2 \cap WD_2} \sigma(w_2)}{|WC_1 \cap WD_1| + |WC_2 \cap WD_2|} \quad (1)$$

où : $WC_i, i \in \{1, 2\}$ est le vocabulaire en langue $\mathscr{L}_i$ associé au corpus C_i ; WD_i est l'ensemble des entrées en langue $\mathscr{L}_i$ du dictionnaire bilingue utilisé présentes dans WC_i ; $\sigma(w_i)$ est une fonction indicatrice qui prend la valeur 1 si au moins une traduction de l'entrée lexicale $w_i \in WC_i$ en langue $\mathscr{L}_i$ existe dans le vocabulaire associé au corpus de l'autre langue, 0 sinon.

2.2 Enrichissement de la mesure LG

La mesure LG ne prend ni en compte le nombre d'occurrences des entrées lexicales dans les documents ni leurs nombres de traductions. Nous proposons ci-après deux variantes de la mesure LG qui font intervenir explicitement ces deux grandeurs en conjecturant que leur prise en compte produira dans certaines situations un effet positif.

2.2.1 Première variante : Cmp_{VA_1}

Cette première variante met en exergue de manière symétrique entre langue cible et langue source les trois éléments suivants : le nombre d'occurrences des entrées lexicales w pris dans le vocabulaire du corpus de la langue source, le nombre de leurs traductions dans le dictionnaire bilingue et la présence d'au moins une de leurs traductions dans le vocabulaire du corpus de la langue cible.

$$Cmp_{VA_1} = 1/2 \cdot (Cmp_{1,2}(C_1, C_2) + Cmp_{2,1}(C_1, C_2)) \tag{2}$$

où :

$$\begin{aligned} Cmp_{1,2}(C_1, C_2) &= \frac{\sum_{w_1 \in WC_1 \cap WD_1} \left(\frac{tf(w_1, C_1)}{\tau(w_1, WD_1)} \cdot \sigma(w_1) \right)}{\sum_{w_1 \in WC_1 \cap WD_1} \left(\frac{tf(w_1, C_1)}{\tau(w_1, WD_1)} \right)} \\ Cmp_{2,1}(C_1, C_2) &= \frac{\sum_{w_2 \in WC_2 \cap WD_2} \left(\frac{tf(w_2, C_2)}{\tau(w_2, WD_2)} \cdot \sigma(w_2) \right)}{\sum_{w_2 \in WC_2 \cap WD_2} \left(\frac{tf(w_2, C_2)}{\tau(w_2, WD_2)} \right)} \end{aligned} \tag{3}$$

avec $tf(w_i, C_i)$ le nombre d'occurrences de l'entrée lexicale w_i dans le corpus C_i de la langue $i \in \{1, 2\}$; $\tau(w_i, WD_i)$ le nombre de traductions de l'entrée lexicale w_i du corpus C_i dans le dictionnaire WD_i. $\sigma(w_i)$ est défini comme précédemment.

2.2.2 Deuxième variante : Cmp_{VA_2}

Cette deuxième variante est très proche de la précédente, elle se distingue essentiellement sur la manière de symétriser la mesure.

$$Cmp_{VA_2}(C_1,C_2) = \frac{\sum_{w_1 \in WC_1 \cap WD_1} \left(\frac{tf(w_1,C_1)}{\tau(w_1,WD_1)} \cdot \sigma(w_1) \right) + \sum_{w_2 \in WC_2 \cap WD_2} \left(\frac{tf(w_2,C_2)}{\tau(w_2,WD_2)} \cdot \sigma(w_2) \right)}{\sum_{w_1 \in WC_1 \cap WD_1} \left(\frac{tf(w_1,C_1)}{\tau(w_1,WD_1)} \right) + \sum_{w_2 \in WC_2 \cap WD_2} \left(\frac{tf(w_2,C_2)}{\tau(w_2,WD_2)} \right)} \quad (4)$$

où $tf(w_i,C_i)$, $\tau(w_i,WD_i)$ et $\sigma(w_i)$ ont la même signification que précédemment.

3 Protocole d'évaluation

Nos expérimentations se sont focalisées sur les langues Anglaise et Française et suivent globalement le protocole proposé dans (Li et Gaussier, 2010). Ce protocole est construit sur le principe d'une dégradation progressive d'un corpus parallèle par remplacement déterministe par blocs de lignes. Nous avons complété ce protocole en développant une approche non-déterministe pour le remplacement des blocs afin d'évaluer l'impact de la procédure de remplacement des blocs sur la qualité observée des mesures.

3.1 Mesure d'évaluation

3.1.1 Référence empirique étalon

La référence empirique est construite sur la base du pourcentage de dégradation du corpus Europarl. Par exemple, si nous considérons 100 lignes par bloc, pour chaque bloc et pour chaque test, nous obtenons un vecteur de 101 valeurs (en partant de 0% de remplacement pour aboutir à 100% de remplacements). Nous obtenons ainsi une mesure de référence empirique, dite étalon, caractérisée par un vecteur (0%, 1%, 2%...100%) de $N = 101$ coordonnées.

3.1.2 Comparaison d'une mesure de comparabilité à la référence empirique

Pour établir le degré d'adéquation/d'inadéquation d'une mesure à la référence empirique, nous utilisons le coefficient de corrélation de Pearson. Celui-ci estime le degré de corrélation

entre une mesure de comparabilité X et la référence empirique Y de la manière suivante :

$$r_p = \frac{\sum_{n=1}^{N}(X_n - \bar{X}) \cdot (Y_n - \bar{Y})}{\sqrt{\sum_{n=1}^{N}(X_n - \bar{X})^2}\sqrt{\sum_{n=1}^{N}(Y_n - \bar{Y})^2}} \quad (5)$$

Parmi d'autres estimateurs de corrélation, le coefficient de corrélation de Pearson est en général utilisé lorsque les variables X et Y sont supposées suivre des lois normales. En l'absence de contre indication particulière ce coefficient nous semble constituer ici un compromis acceptable.

3.1.3 Taux de couverture

Les taux de couverture du dictionnaire et des corpus sont des paramètres qui influencent grandement les mesures de comparabilité. Nous les définissons de la manière suivante :

- on définit le taux de couverture d'un dictionnaire D vis à vis du vocabulaire V associé à un corpus (i.e. ici à un bloc) par la quantité $T_D = \frac{|V \cap D|}{|V|}$.
- on définit le taux de couverture d'un vocabulaire V associé à un corpus (i.e. à un bloc) vis à vis d'un dictionnaire D par la quantité $T_V = \frac{|V \cap D|}{|D|}$.

3.2 Prétraitements et principes d'évaluation

3.2.1 Dictionnaires bilingues utilisés

Nous avons exploité deux dictionnaires bilingues dans le cadre de cette étude pour évaluer l'impact du dictionnaire de sa couverture sur les mesures de comparabilité.

Le premier dictionnaire référencé sous l'intitulé *fullDicText* est un dictionnaire propriétaire qui contient 74921 paires d'entrées lexicales français/anglais, se décomposant en 32767 d'entrées lexicales en langue anglaise, et 27511 d'entrées lexicales en langue française.

Le deuxième dictionnaire référencé sous l'intitulé *dicElra*, et disponible sous la référence ELRA-M0033, contient 243580 paires d'entrées lexicales en langues française et anglaise, se décomposant en 110541 entrées lexicales en langue anglaise et 109196 entrées lexicales en langue française.

3.2.2 Prétraitements

Nous disposons de deux corpus : un corpus parallèle «français-anglais Europarl corpus» (Koehn, 2005) et un corpus anglais «Associated Press corpus : AP». Ces corpus sont lemmati-

sés en exploitant le TreeTagger (Schmid, 1994) (Schmid, 2009) puis segmentés en phrases (une phrase par ligne). A l'issue de ce prétraitement, nous disposons ainsi de trois documents contenant chacun plusieurs millions de lignes : un document parallèle français EPF, un document parallèle anglais EPE et un document anglais AP.

3.2.3 Principes d'évaluation

En suivant les travaux de (Li et Gaussier, 2010), nous partitionnons le corpus parallèle Europarl en sélectionnant un nombre variable de lignes : 1000 lignes, 10000 lignes, 100000 lignes et 1428000 lignes (ce qui correspond à l'intégralité du corpus Europarl). Chaque élément de la partition obtenue est ensuite divisée en 10 blocs, chaque bloc contenant le même nombre de lignes (100 lignes, 1000 lignes, 10000 lignes, et 142800 lignes). Nous calculons ensuite les mesures au niveau des blocs alignés.

Nous proposons deux séries d'expérience qui se distinguent par le mode de remplacement : déterministe ou aléatoire. Pour chacune de ces séries, trois tests différents sont effectués selon les principes décrits ci-après. L'évaluation des mesures de comparabilité consiste à quantifier la corrélation entre leur décroissance observée et la décroissance attendue d'une mesure *empirique* quantifiant le degré de dégradation du corpus parallèle initial.

3.2.4 Remplacement déterministe

Pour le premier test, nous construisons les corpus référencés par *GAd* en remplaçant par permutation un certain nombre de lignes issues d'un bloc (le nombre de lignes est fonction du pourcentage de dégradation du corpus parallèle $0\%, 1\% \cdots 100\%$) par le même nombre de lignes issues d'un autre bloc. La permutation des blocs est prédéfinie, par exemple :le bloc 1 <-> le bloc 6, bloc 2 <-> le bloc 7, etc.

Pour le deuxième test, nous construisons les corpus référencés par *corpus GBd*, en remplaçant certaines lignes issues d'un bloc (le nombre de lignes est fonction du pourcentage de dégradation du corpus parallèle souhaité) par le même nombre de lignes extraites du document *AP*.

Pour le troisième test, nous construisons les corpus référencés par *corpus GCd*, en remplaçant toutes les lignes d'un bloc par toutes les lignes d'un autre bloc, c'est-à-dire par exemple, le bloc 1 devient le bloc 6 et le bloc 2 devient le bloc 7, etc. A ce stade, et dans chaque bloc, un certain nombre de lignes (fonction du pourcentage de dégradation du corpus parallèle souhaité) sont remplacées par un même nombre de lignes extraites du fichier *AP*.

3.2.5 Remplacement aléatoire

Pour le premier test, nous construisons les corpus référencés par *corpus GAa* en remplaçant aléatoirement selon une loi uniforme un certain nombre de lignes, en fonction du pourcentage de dégradation du corpus parallèle souhaité, par le même nombre de lignes extraites (sans remise pour garantir que les remplacements concernent systématiquement des lignes différentes) du reste des lignes non exploitées du corpus parallèle.

Pour le deuxième test, nous construisons les corpus référencés par *corpus GBa* en remplaçant aléatoirement selon une loi uniforme un certain nombre de lignes, en fonction du pourcentage de dégradation du corpus parallèle souhaité, par le même nombre de lignes extraites du document *AP*, en supprimant les lignes de remplacement déjà exploitées du document *AP*.

Pour le troisième test, nous construisons le corpus référencé par *corpus GCa*, en remplaçant d'abord toutes les lignes d'un bloc par le même nombre de lignes issues du complément du bloc dans l'ensemble des lignes du corpus Europarl (sans remplacement). Ensuite, au sein de chaque bloc, nous effectuons le remplacement aléatoire selon une loi uniforme d'un nombre de lignes donné (qui dépend du pourcentage de dégradation du corpus Europarl souhaité) par le même nombre de lignes extraites du corpus *AP* sans remplacement.

Ainsi, pour les deux séries de trois tests, le degré de comparabilité moyen décroît, en principe, de $GA_{d|a}$ à $GC_{d|a}$, en passant par $GB_{d|a}$.

4 Expérimentations

4.1 Influence de la taille des blocs sur les corrélations moyennes

Nous étudions ici les corrélations moyennes et leurs écarts-types entre les mesures de comparabilité et la référence empirique lorsque la taille des blocs exprimée en nombre de lignes varie dans l'ensemble $\{10^2, 10^3, 10^4, 10^5\}$.

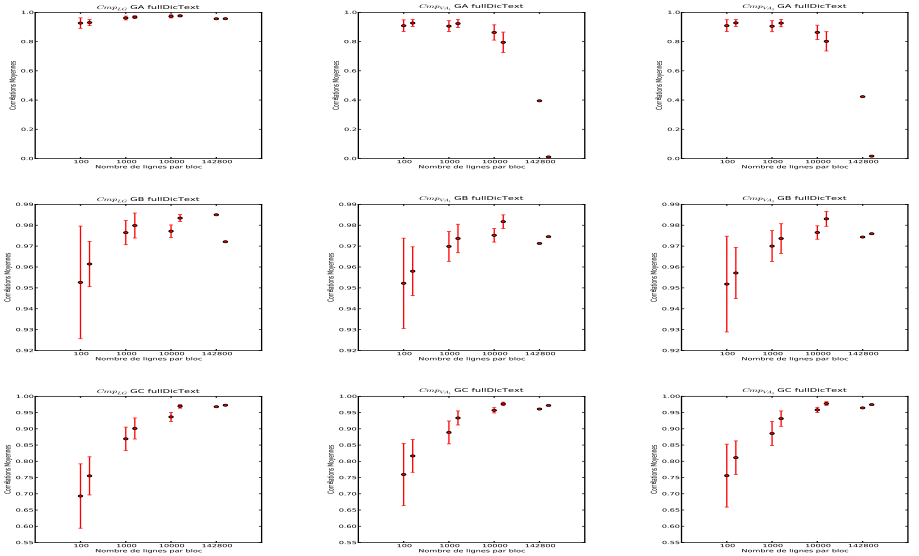


FIGURE 1 – Influence de la taille des blocs de corpus sur les corrélations moyennes des mesures vis-à-vis de la référence empirique étalon pour le dictionnaire bilingue *fullDicText*. Les deux modes de remplacement sont représentés pour chaque taille de bloc avec un léger décalage : déterministe à gauche et aléatoire à droite

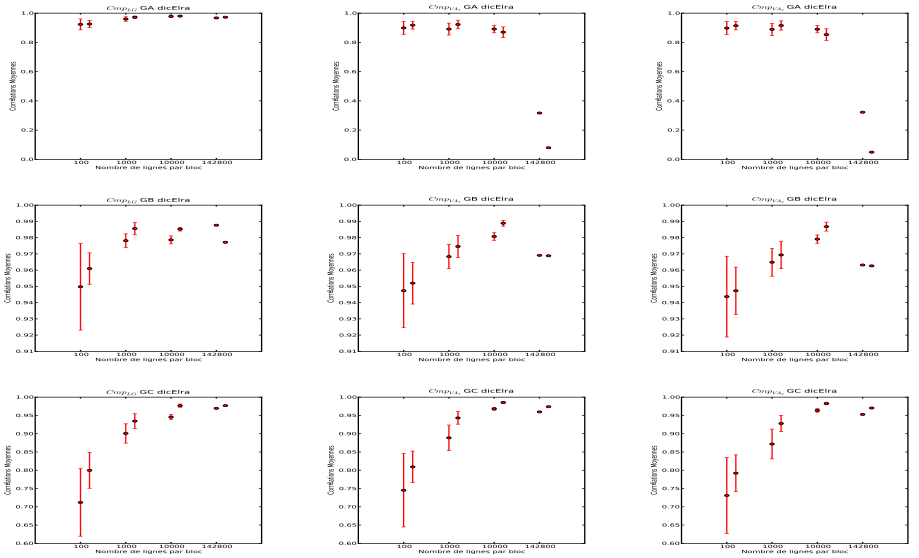


FIGURE 2 – Influence de la taille des blocs de corpus sur les corrélations moyennes des mesures vis-à-vis de la référence empirique étalon pour le dictionnaire bilingue *Elra*. Les deux modes de remplacement sont représentés pour chaque taille de bloc avec un léger décalage : déterministe à gauche et aléatoire à droite

Les figures 1 et 2 montrent, pour les deux modes de remplacement, que la mesure Cmp_{LG} est plus en adéquation avec la référence empirique au sens du coefficient de corrélation de Pearson sur les expériences *GA* que ses variantes Cmp_{VA_1} et Cmp_{VA_2}, en particulier pour des tailles de blocs importantes. Pour les expériences *GB*, les trois mesures atteignent quasiment le même niveau de corrélation vis-à-vis de la référence empirique. Enfin, sur les expériences *GC*, les deux variantes Cmp_{VA_1} et Cmp_{VA_2} semblent être légèrement plus robustes que Cmp_{LG}, principalement pour des tailles de bloc petites. Les deux dictionnaires bilingues utilisés conduisent à des résultats très voisins. Par contre, la procédure de remplacement aléatoire semble améliorer pour toutes les mesures et pour les deux dictionnaires la corrélation avec la référence empirique étalon, tant en moyenne qu'en écart type.

4.2 Influence des taux de couverture sur les corrélations moyennes des mesures avec la référence empirique

Nous étudions ici l'influence des taux de couverture (des dictionnaires et des vocabulaires en faisant varier la taille des blocs) sur les corrélations moyennes vis-à-vis de la référence empirique étalon obtenues sur la base des corpus dégradés par remplacement déterministe ou aléatoire, ceci pour les trois mesures Cmp_{LG}, Cmp_{VA_1} et Cmp_{VA_2}. Les figures 3 et 4 présentent ces corrélations moyennes pour les deux dictionnaires *fullDicText* et *dicElra* et pour les deux modes de remplacement, aléatoire et déterministe.

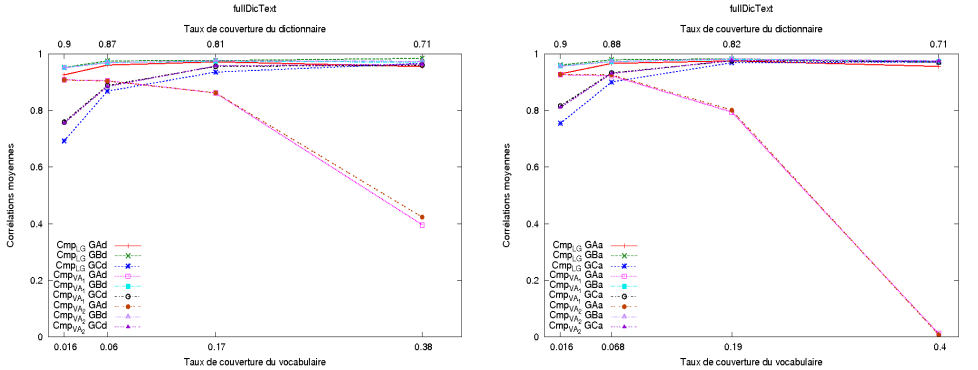


FIGURE 3 – Influence du taux de couverture sur les corrélations moyennes des mesures vis-à-vis de la référence empirique étalon pour le dictionnaire *fullDicText*, à gauche pour les corpus dégradés par remplacement déterministe, à droite pour les corpus dégradés par remplacement aléatoire

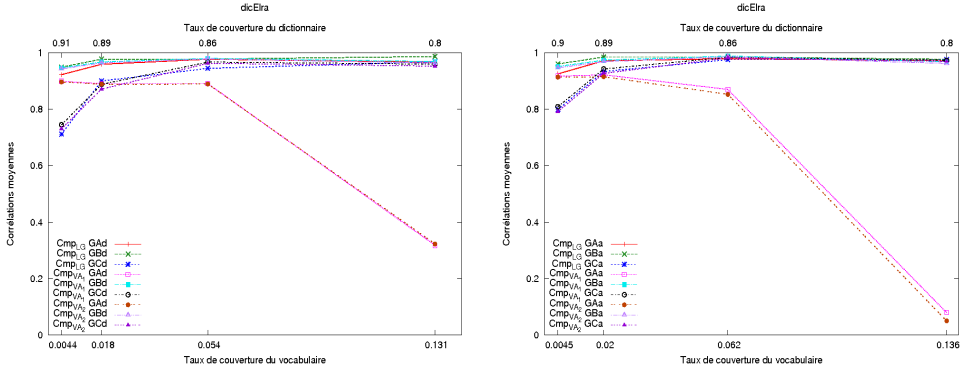


FIGURE 4 – Influence du taux de couverture sur les corrélations moyennes des mesures vis-à-vis de la référence empirique étalon pour le dictionnaire *dicElra*, à gauche pour les corpus dégradés par remplacement déterministe, à droite pour les corpus dégradés par remplacement aléatoire

On constate sur les figures 3 et 4 une meilleure corrélation moyenne pour la mesure Cmp_{LG} sur les corpus *GA*, tandis que les variantes Cmp_{VA_1} et Cmp_{VA_2} voient leurs corrélations s'effondrer sur ce même corpus lorsque le taux de couverture du dictionnaire croit. Sur les corpus *GB*, les trois mesures ont des performances très voisines, tandis que, sur les corpus *GC*, les deux variantes sont un peu mieux corrélées à la référence que la mesure Cmp_{LG}. Nous notons également une légère baisse en corrélation moyenne qui s'observe pour les trois mesures lorsque le taux de couverture du vocabulaire est très faible. Ces résultats sont analogues pour les deux dictionnaires *fullDicText* et *Elra* ainsi que pour les deux modes de remplacement déterministe et aléatoire.

4.3 Capacités des mesures à discriminer les degrés de dégradation du corpus parallèle Europarl

Afin de quantifier la capacité des mesures à discriminer les différents niveaux de dégradation du corpus parallèle Europarl au fur et à mesure des remplacements, que ceux-ci soient déterministes ou aléatoires, nous utilisons la mesure de discrimination suivante :

$$\Delta(i) = \frac{|\sigma_i + \sigma_{i+1} + 2 \cdot (m_i - \sigma_i/2 - (m_{i+1} + \sigma_{i+1}/2))|}{\sigma_i + \sigma_{i+1}} = \frac{2 \cdot |m_i - m_{i+1}|}{\sigma_i + \sigma_{i+1}} \quad (6)$$

où m_i et σ_i sont les moyennes et écarts types des valeurs de comparabilité associées aux niveaux (de 0%, 1%, $\cdots$100%) de dégradation du corpus Europarl indexés par $i \in \{1, \cdots, 101\}$. En pratique, on observe que $\forall i, m_i >= m_{i+1}$ et la valeur absolue n'est pas requise. $\Delta(i) \in [0, \infty[$ est d'autant plus grande que l'écart entre les comparabilités moyennes successives est grand et que la somme des écarts types associés est faible. Ainsi, plus la fonction $\Delta(i)$ est élevée, mieux le niveau i de dégradation du corpus est discriminé par la mesure de comparabilité.

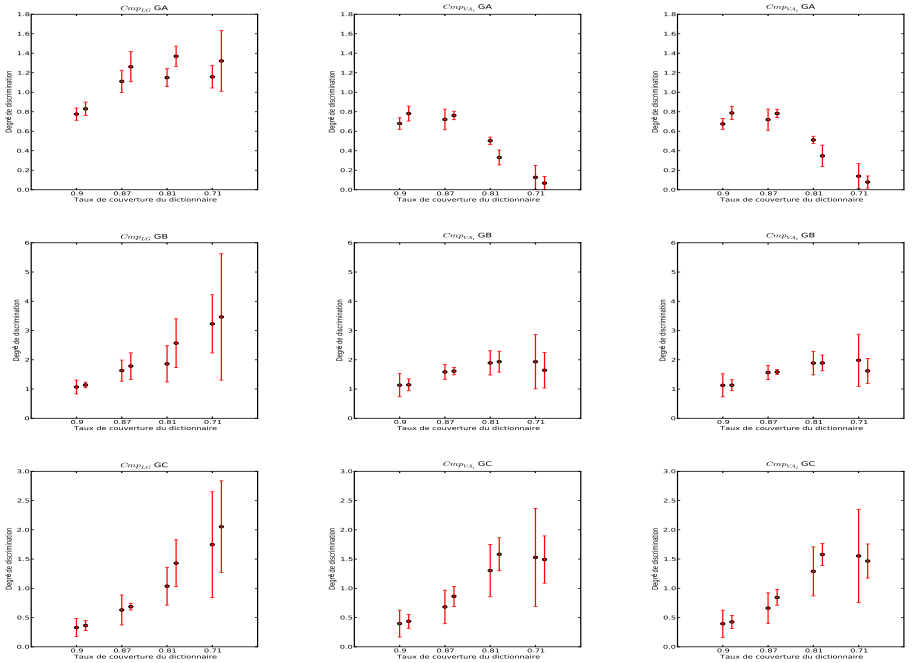


FIGURE 5 – Capacité des mesures de comparabilité à discriminer les degrés de dégradation du corpus Europarl : moyennes et écarts-types de $\Delta(.)$ en fonction des taux de couverture du dictionnaire *fullDicText* exploité sur les corpus produits par remplacements déterministe (décalages à gauche) et aléatoire (décalages à droite).

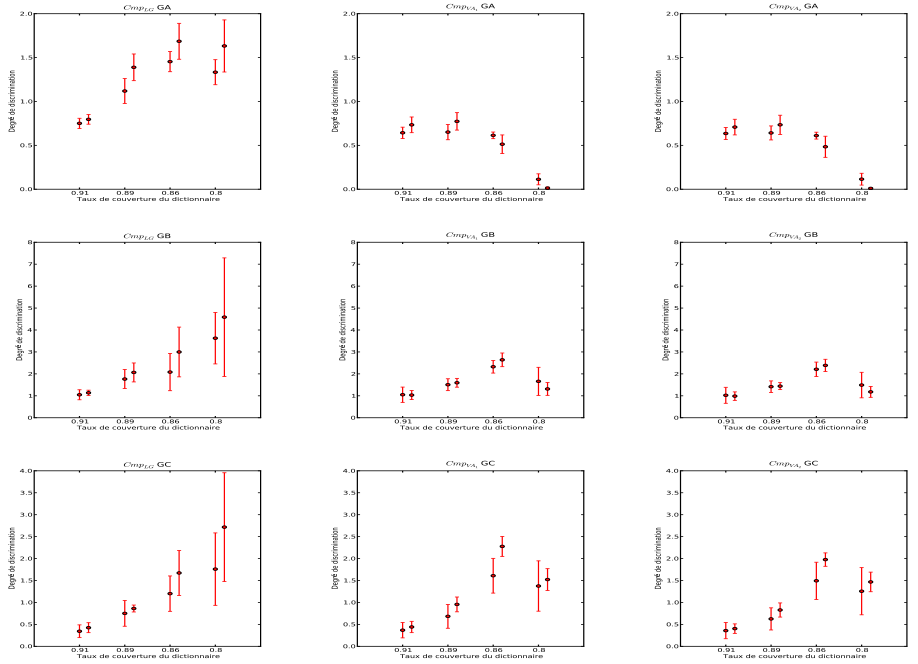


FIGURE 6 – Capacité des mesures de comparabilité à discriminer les degrés de dégradation du corpus Europarl : moyennes et écarts-types de $\Delta(.)$ en fonction des taux de couverture du dictionnaire *dicElra* exploité sur les corpus produits par remplacements déterministe (décalages à gauche) et aléatoire (décalages à droite).

Les figures 5 et 6 présentent pour les trois mesures Cmp_{LG}, Cmp_{VA_1} et Cmp_{VA_2}, sur les trois types de corpus (*GA*, *GB* et *GC*) la valeur moyenne et l'écart type de la mesure de discrimination Δ en fonction du taux de couverture des dictionnaire *fullDicText* et *dicElra* respectivement. Ici également, on constate que les variantes Cmp_{VA_1} et Cmp_{VA_2} sont moins discriminantes que la mesure Cmp_{LG} sur les corpus *GA* surtout pour les taux de couverture faible. Sur les corpus *GB* et *GC*, les mesures ont des niveaux de corrélation très voisins, surtout pour les taux de couverture les plus élevés du dictionnaire. Enfin, Sur les corpus *GC*, les variantes semblent légèrement plus robustes, notamment pour les taux de couverture élevés du dictionnaire. A noter que la capacité de discrimination moyenne augmente lorsque le taux de couverture du dictionnaire diminue dans la plupart des cas, mais sa variance augmente également en proportion également dans la plupart des cas.

5 Analyse et conclusions

Les résultats obtenus montrent que la mesure Cmp_{LG} et ses variantes Cmp_{VA_1}, Cmp_{VA_2} sont relativement voisines du point de vue de leur corrélation vis-à-vis de la mesure empirique étalon définie dans le contexte du protocole d'évaluation mis en œuvre. Il ressort néanmoins

que la mesure Cmp_{LG} est bien mieux corrélée à la mesure étalon sur les corpus les plus proches du corpus parallèle initial (Europarl) *GAd* et *GAa*, tandis que les variantes Cmp_{VA_1}, Cmp_{VA_2} sont légèrement plus robustes lorsque les mesures sont confrontées aux corpus *GCd* et *GCa*, les plus éloignés du corpus Europarl et sans doute les plus proches des corpus *bruités* tels que ceux constitués à partir de données collectées sur le Web par exemple. Sur les corpus intermédiaires *GBd* et *GBa* les trois mesures atteignent des niveaux de corrélation comparables vis-à-vis de la mesure empirique étalon.

Les dictionnaires ont un léger effet sur la corrélation entre nos deux variantes de comparabilité et la mesure empirique étalon : pour le dictionnaire *fullDicText*, Cmp_{VA_2} est légèrement mieux corrélée à la mesure étalon, tandis que pour le dictionnaire *dicElra*, c'est la variante Cmp_{VA_1} qui semble mieux corrélée.

Les degrés de corrélation de ces mesures augmentent lorsque le nombre de lignes par bloc augmente, en particulier pour le corpus *GC* (augmentation de plus de 20% entre la configuration 100 lignes par bloc et la configuration 142800 lignes par bloc). Par exemple, pour deux documents d'environ 100 lignes chacun, si la valeur de comparabilité est supérieure à 0.7, les deux documents sont probablement très comparables et pour deux documents de plus de 1000 lignes chacun, si la valeur de comparaison est supérieure à 0.8, les deux documents sont probablement comparables au même degré que les précédents. A l'appui de ce résultat, nous pouvons espérer proposer une référence raisonnablement stable pour la comparabilité des documents en fonction de leur nombre de phrases afin de juger si les documents sont suffisamment comparables ou non pour la tâche considérée.

Par ailleurs, les capacités des mesures à discriminer les niveaux successifs de dégradation du corpus parallèle que nous proposons est également un critère de comparaison intéressant nous semble-t-il. Sur ce critère, les tendances précédemment évoquées restent en vigueur. La mesure Cmp_{LG} se comporte mieux sur les corpus *GA* tandis que les variantes Cmp_{VA_1} et Cmp_{VA_2} semblent plus discriminantes sur les corpus *GC* et peut être également *GB* compte tenu des variances plus faibles observées sur ce critère pour les deux variantes.

Les modes de remplacement aléatoire ou déterministe semblent avoir un impact assez significatif au vu des résultats. Sur le corpus Europarl, le protocole déterministe de dégradation du remplacement proposé par (Li et Gaussier, 2010) engendre, en général, une baisse en moyenne des corrélations des trois mesures évaluées ainsi qu'un accroissement des écarts types, surtout sur les corpus s'éloignant du corpus parallèle Europarl (i.e. *GB* et *GC*). Cela amène à privilégier le mode de remplacement aléatoire par rapport au mode déterministe.

En matière de perspective, d'une part, nous allons essayer d'améliorer la précision lorsque le taux de couverture du dictionnaire est faible ; et d'autre part, nous allons exploiter et évaluer ces mesures de comparabilité dans le cadre d'expérimentations portant sur des réelles, en particulier sur des tâches de *bi-classification* et de *bi-clustering* de données thématiques bilingues.

Remerciements

Ces travaux ont été partiellement financés dans le cadre du projet ANR-08-CORD-009 METRICC.

Références

DÉJEAN, H. et GAUSSIER, E. (2002). Une nouvelle approche a l'extraction de lexiques bilingues à partir de corpus comparables. *Lexicometrica*, Numéro spécial, corpus alignés:1–22.

FUNG, P. et YEE, L. Y. (1998). An ir approach for translating new words from nonparallel, comparable texts. *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 414–420, Stroudsburg, PA, USA. Association for Computational Linguistics.

KOEHN, P. (2005). Europarl : A Parallel Corpus for Statistical Machine Translation. *In Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

LI, B. et GAUSSIER, E. (2010). Improving corpus comparability for bilingual lexicon extraction from comparable corpora. *In COLING*, pages 644–652.

MUNTEANU, D. S., FRASER, A. et MARCU, D. (2004). Improved machine translation performance via parallel sentence extraction from comparable corpora. *In HLT-NAACL*, pages 265–272.

SCHMID, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. *In Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

SCHMID, H. (2009). TreeTagger, www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

Inférence grammaticale guidée par clustering

Noémie-Fleur Sandillon-Rezer
CNRS, Esplanade des Arts et Métiers, 33402 Talence
LaBRI, 351 Cours de la Libération, 33405 Talence
nfsr@labri.fr

RÉSUMÉ

Dans cet article, nous nous focalisons sur la manière d'utiliser du clustering hiérarchique pour apprendre une grammaire AB à partir d'arbres de dérivation partiels. Nous décrirons brièvement les grammaires AB ainsi que les arbres de dérivation dont nous nous servons comme entrée pour l'algorithme, puis la manière dont nous extrayons les informations des corpus arborés pour l'étape de clustering. L'algorithme d'unification, dont le pivot est le cluster, sera décrit et les résultats analysés en détails.

ABSTRACT

Clustering for categorial grammar induction

In this article, we describe the way we use hierarchical clustering to learn an AB grammar from partial derivation trees. We describe AB grammars and the derivation trees we use as input for the clustering, then the way we extract information from Treebanks for the clustering. The unification algorithm, based on the information extracted from our cluster, will be explained and the results discussed.

MOTS-CLÉS : grammaires catégorielles, clustering hiérarchique, inférence grammaticale.

KEYWORDS: categorial grammars, hierarchical clustering, grammatical inference.

1 Introduction

Le but de cet article est de présenter une nouvelle méthode d'inférence grammaticale. En effet, nous utilisons en entrée de notre algorithme des arbres de dérivations d'une grammaire AB partiellement remplis, et l'algorithme est ensuite guidé par le clustering pour savoir quelles variables vont être unifiées. L'idée de base est que les mots qui sont dans des contextes similaires doivent avoir des types similaires.

L'inférence grammaticale appliquée aux grammaires catégorielles peut être classée en trois catégories distinctes, en fonction des méthodes employées et des structures d'entrée.

La méthode décrite par Adriaans (Trautwein *et al.*, 2000; Adriaans, 1999) a pour point de départ des phrases sans structure. Bien qu'elle fonctionne dans la plupart des cas, ce genre de méthode d'inférence rencontre des problèmes avec des phrases permettant plusieurs analyses syntaxiques qu'on ne peut pas distinguer et qui sont fondées seulement sur la chaîne des mots

-par exemple l'attachement des syntagmes prépositionnels. Il semble donc logique, étant donné que cette information est généralement annotée dans des corpus arborés, de l'utiliser lors de l'apprentissage.

Des structures partielles sont utilisées dans les méthodes de Buszkowski et Penn (Buszkowski et Penn, 1990) ou Kanazawa (Kanazawa, 1998). Ces méthodes sont clairement dans la lignée du paradigme de Gold (Gold, 1967). Les structures d'entrée sont décrites ultérieurement, section 3. La sortie de ces algorithmes donne soit une grammaire rigide [1] (algorithme de Buszkowski et Penn) soit une grammaire k-valuée [2] (algorithme de Kanazawa). Une grammaire rigide n'est pas représentative d'un langage naturel, et dès $k \geq 2$, l'unification devient un problème NP-dur (Costa-Florêncio, 2001). Outre ce fait, comme k n'est pas connu par avance, il convient de trouver la valeur optimale, ce qui est particulièrement complexe. On doit alors appliquer une recherche dichotomique pour trouver k, partant du principe que celui-ci se situe au départ entre 1 et ∞. L'idée est de prendre une valeur arbitraire qui semble raisonnable, de tester. Si cela fonctionne, on divise la valeur par deux et on tente à nouveau, sinon on la multiplie par deux en partant du principe que la bonne valeur devra être supérieure ou égale à $k + 1$. Il est à noter, cependant, que même une grammaire 2-valuée ne peut pas représenter une langue naturelle : l'expérience avec les grammaires extraites montre que le nombre maximum de types par mot est grand, et de nombreux mots fréquents (déterminants, conjonctions de coordinations) possèdent plus de quarante types (Hockenmaier et Steedman, 2007; Sandillon-Rezer et Moot, 2011).

Enfin, les méthodes utilisent des structures totalement définies, comme celle d'Hockenmaier (Hockenmaier, 2003), qui construit une grammaire catégorielle combinatoire, ou notre méthode (Sandillon-Rezer et Moot, 2011), qui utilise un transducteur d'arbres généralisé pour transformer des arbres syntaxiques en arbres de dérivation d'une grammaire AB (Lambek, 1958). C'est la sortie de notre transducteur qui nous servira de standard d'évaluation ainsi que d'entrée après modification des arbres pour notre algorithme d'inférence grammaticale.

Dans cet article nous combinons les méthodes du second type avec des structures partielles (agrémentées de certaines informations provenant des corpus) et du clustering. Nous évaluons à la fois la complexité du problème et la qualité du lexique obtenu. Le clustering est effectué en utilisant une mesure de similarité fondée sur le contexte local du mot, qui est directement extrait des arbres syntaxiques.

Les arbres de dérivation sont extraits de corpus annotés. Nous utilisons deux corpus différents en guise de base : le corpus de Paris VII (Abeillé *et al.*, 2003) et Sequoia (Candito et Seddah, 2012). Les deux corpus sont annotés de manière syntaxique par les mêmes protocoles d'annotation (Abeillé et Clément, 2003). Les principales différences entre les deux corpus résident dans le nombre de phrases et l'origine de celles-ci. Le corpus de Paris VII est composé de 12351 phrases qui proviennent d'une sélection d'articles du journal *Le Monde*, et Sequoia est composé de 3204 phrases venant de différents horizons, comme Wikipedia, le journal *L'Est Républicain* ou encore des notices médicales. La figure 1 donne un exemple d'arbre syntaxique. Les noeuds pré-terminaux contiennent les POS-tag [3] du mot, les autres noeuds internes contiennent le type syntagmatique du sous-arbre et les feuilles représentent les mots de la phrase.

Etant donné que le format des annotations ne correspond pas aux arbres de dérivation d'une grammaire AB, nous utilisons le transducteur d'arbres généralisé pour transformer les arbres du

1. Une grammaire catégorielle rigide force les mots du lexique à n'avoir qu'un seul type.
2. Une grammaire k-valuée permet aux mots d'un lexique d'avoir k types au maximum.
3. les annotations *parties du discours*

corpus de Paris VII et de Sequoia en arbres de dérivation.

Nous commencerons par décrire la grammaire AB générée par le transducteur généralisé, ensuite nous rappellerons le principe général de l'algorithme d'unification pour les grammaires AB et décrirons celui que nous utilisons. Dans la section quatre, nous décrirons le format de vecteurs utilisés pour l'étape de clustering. L'évaluation de notre méthode suivra, ainsi qu'une discussion sur les extensions possible de ce travail.

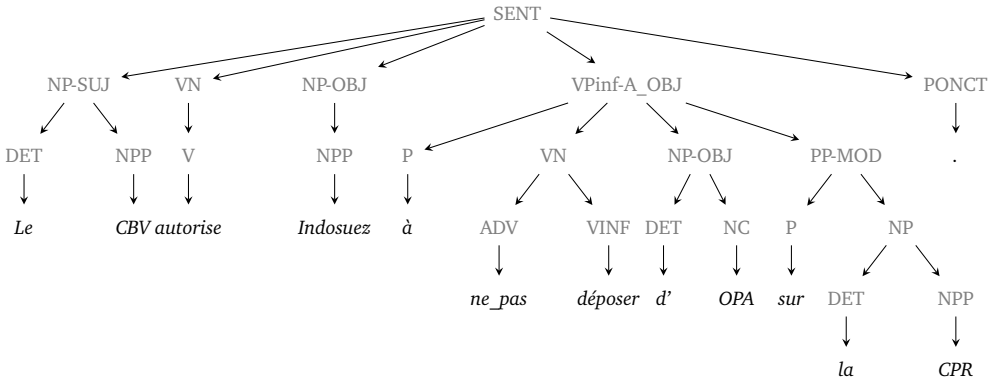


FIGURE 1 – Exemple d'arbre syntaxique du corpus de Paris VII.

2 Arbres de dérivation d'une grammaire AB

Les grammaires AB ont été définies séparément par Ajdukiewicz (Ajdukiewicz, 1935) et Bar-Hillel (Bar-Hillel, 1964) à partir du coeur des grammaires catégorielles et sont à présent considérées comme une sous-partie du calcul de Lambek (Lambek, 1958) et des grammaires catégorielles combinatoires [4] (Hockenmaier et Steedman, 2007). Les grammaires AB ont seulement deux règles d'éliminations, comme montré tableau 1.

$$\frac{A/B \quad B}{A}\ [/E] \qquad \frac{B \quad B\backslash A}{A}\ [\backslash E]$$

TABLE 1 – The elimination rules for AB grammar

Les arbres de dérivation d'une grammaire AB représentent l'application successive des règles d'éliminations.

Notre transducteur généralisé, qui correspond à une version modifiée d'un transducteur descendant, transforme les arbres syntaxiques des deux corpus en dérivations d'une grammaire AB. La figure 2 montre un exemple de sortie du transducteur correspondant à l'arbre syntaxique de la

4. Il est à noter cependant que nous suivons la convention déterminée par Lambek d'avoir toujours la catégorie qui sert d'argument sous le slash.

figure 1. Moins de 1.650 règles de transduction sont nécessaires pour convertir 92% du corpus de Paris VII (93% de Sequoia).

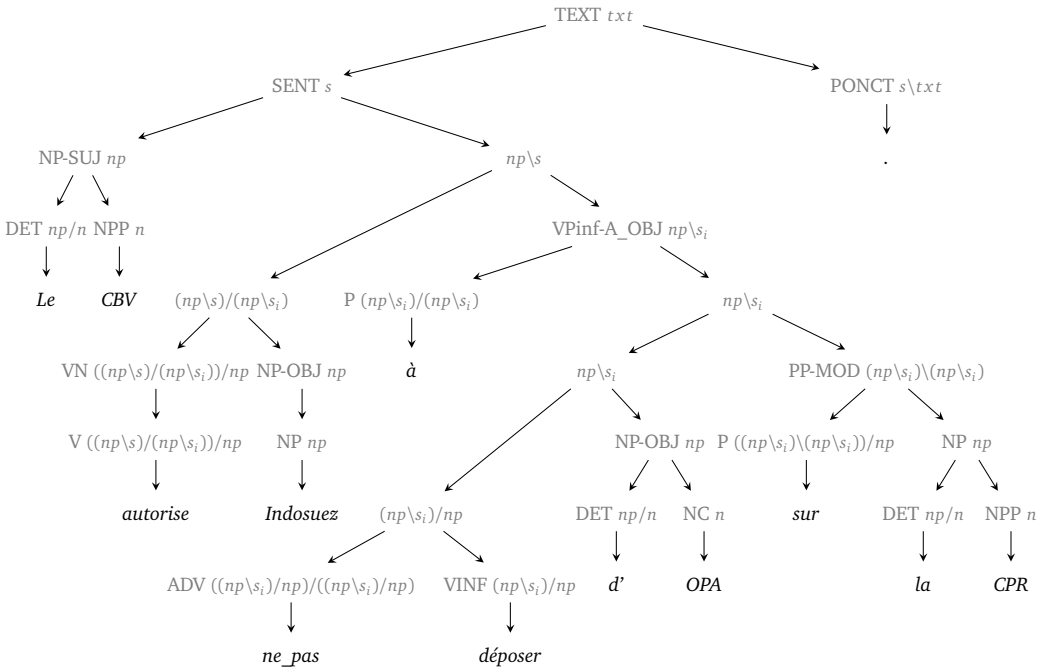


FIGURE 2 – Sortie du transducteur. Les informations provenant de l'arbre syntaxique sont toujours présentes. Il est à noter que sur la sortie réelle, les types sont aussi présents dans les feuilles ; ils sont hérités des noeuds pré-terminaux.

Le transducteur utilise quelques types atomiques pour l'étape de transduction : *np, n, s, txt, pp, cs,* cl_r. Cela correspond respectivement, à : un syntagme nominal, un nom commun, une phrase, un "texte" (une phrase avec une ponctuation finale), un syntagme prépositionnel, une clause subordonnée et un clitique réflexif. En plus, nous utilisons les types $np\backslash s_p$ et $np\backslash s_i$ pour les syntagmes participiaux et infinitivaux. Cependant, le transducteur est relativement modulaire. Chacun peut créer un ensemble de règles pour binariser les arbres et le transducteur vérifiera que les arbres sont bien binaires à la sortie et que les types sont cohérents au sens de Ajdukiewicz, c'est à dire qu'on a bien à chaque étape une application d'une des règles d'élimination.

Nous utiliserons ces types pour initialiser nos arbres avant l'étape d'unification ; la description du format d'entrée est effectuée dans la section 3.

3 Inférence grammaticale

Un algorithme d'inférence grammaticale bien connu est celui décrit par Buszkowski et Penn (Buszkowski et Penn, 1990). Pour apprendre une grammaire rigide, cela ne pose pas de problème

(voir algorithme 1) : soit les types du lexique peuvent être unifiés jusqu'à ce qu'il n'y en ait plus qu'un par mot, soit l'algorithme échoue. Pour apprendre une grammaire k-valuée (Kanazawa, 1998) il y a besoin du même format d'entrée, comme montre la figure 3. Le coeur du problème avec les grammaires k-valuées est de décider quels types doivent être unifiés, car la meilleure unification possible ne peut être décidée que d'un point de vue global. C'est pour cela que ce problème d'inférence grammaticale est NP-dur (Costa-Florêncio, 2001) dès que $k \geq 2$. Il est également important de noter que k n'est généralement pas connu d'avance, ce qui complique la résolution de l'algorithme.

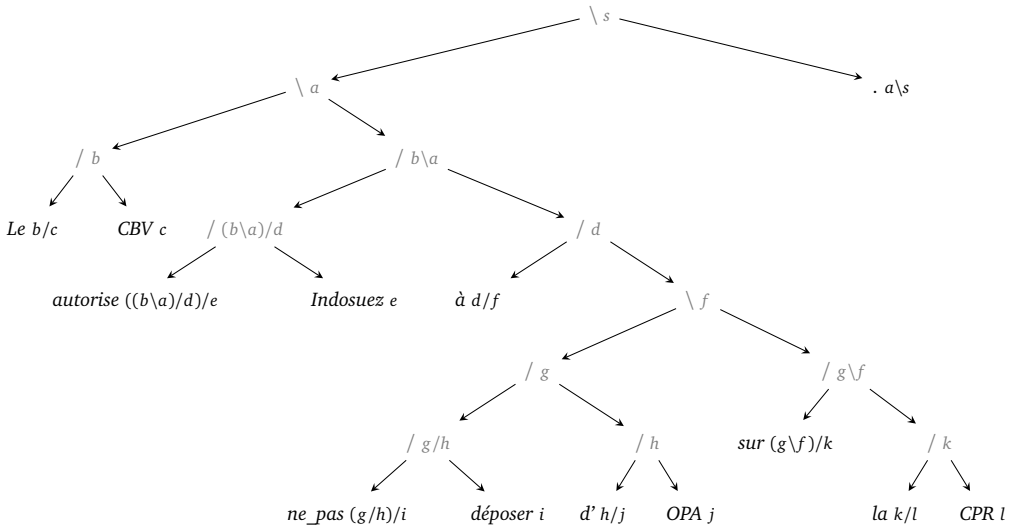


FIGURE 3 – Exemple d'entrée pour les deux algorithmes d'unification.

Données: arbres dont la racine est étiquetée s, les noeuds internes / ou \, et les feuilles avec des variables
Résultat: une grammaire rigide
création d'un lexique de mots contenant toutes les variables qui leurs sont liées ;
tant que *chaque mot a plusieurs types qui lui sont liés* **faire**
  rechercher l'unificateur le plus généraliste, de manière à réduire globalement le nombre de variables liées aux mots;
  si *l'unification n'est pas possible* **alors** l'algorithme échoue;
fin

Algorithm 1: Algorithme d'apprentissage d'une grammaire rigide.

Pour illustrer le problème de l'unification, il suffit de prendre deux phrases du corpus de Paris VII : *Nous avons de plus en plus de monde dans les DOM-TOM* et *Le gouvernement n'avait ni écrit ni choisi cet accord dont nous avons hérité*. Le lexique avant unification est décrit tableau 2. En appliquant l'algorithme de Buszkowski et Penn, le verbe *avons* aura deux types qui se ressemblent : $(d\backslash c)/e$ et $(w\backslash v)/x$. En effet, à chaque fois *avons* prend deux arguments, l'un à sa gauche et l'autre à sa droite. Cependant, nous ne pouvons pas unifier ces deux types, parce que dans le premier cas

l'argument de droite est un groupe nominal et dans le second cas un participe passé ; en outre nous ne souhaitons pas que les deux aient le même type, pour éviter des phrases agrammaticales. Pour ces deux phrases, nous avons donc besoin au minimum d'une grammaire 2-valuée, et *avons* doit avoir deux types différents : $(np\backslash s)/np$ et $(np\backslash s)/(np\backslash s_p)$.

nous	d, w	avons	$(d\backslash c)/e$, $(w\backslash v)/x$	de_plus_en_plus	e/f
de	f/g	monde	g	dans	$(c\backslash a)/h$
les	h/i	DOM-TOM	i	le	l/m
gouvernement	m	n'	$((l\backslash k)/n)/p$	avait	$(o/p)/q$
ni	$q/r, p/s$	écrit	r	cet	n/t
accord	u	dont	$(u\backslash t)/v$	hérité	x
.	$k\backslash j, a\backslash b$				

TABLE 2 – Lexique avant unification

Entre ces algorithmes standards d'inférence grammaticale et le nôtre, il y a deux différences principales.

La première différence réside dans le fait que nous utilisons les informations provenant des annotations du corpus, comme résumé dans la table 3. Les types assignés aux noeuds ont été choisis en fonction de leur fréquence dans le lexique extrait des arbres après transduction. Si le label n'est pas dans la table, le type du noeud sera une variable dans le cas d'un noeud argument. Si le noeud est foncteur, son type sera instancié en même temps que celui de son argument, puisque cette méthode est descendante. Les arbres utilisés en entrée contiennent donc des sous-formules avec des variables libres. L'inférence grammaticale consiste à transformer ces arbres aux types partiellement spécifiés en arbres de dérivation sans variable. Un exemple d'arbre d'entrée est montré figure 4. On remarque que certains mots, même si leur POS-tag n'est pas dans la table 3, ont déjà des types complexes sans variable.

label	**type**	**label**	**type**
TEXT	txt	SENT	s
NP	np	NP-ARG	np
PP	pp	AP-ARG	n\n
CLS	np	CLS-SUJ	np
NC	n	NPP	np
VPP	np\ s_p	VINF	np \ s_i

TABLE 3 – Extrait de la liste des types assignés aux noeuds lorsque ceux-ci ont le bon label, si et seulement s'ils sont arguments et non foncteurs au niveau des dérivations d'une grammaire AB.

La seconde différence est l'utilisation de clusters pour guider l'étape d'unification. Nous avons pris le parti d'utiliser un algorithme de clustering hiérarchique pour ce faire. Un cluster peut aussi bien regrouper un ensemble de cluster que des mots. Chaque cluster est associé à une hauteur qui représente la similarité entre les données qu'il regroupe. Ainsi, les clusters de hauteur zéro regroupent les mots dont les vecteurs sont identiques, et les clusters de hauteur plus importante regroupent aussi bien des mots que d'autres clusters qui leurs sont proches, comme montré figure 5. Nous unifions les clusters par hauteur croissante, ce qui nous permet d'unifier par ordre de

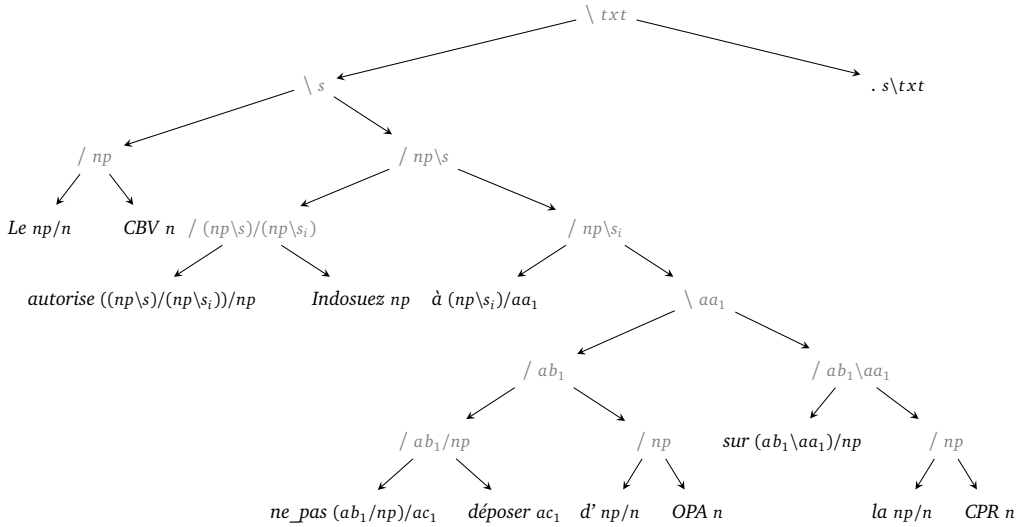


FIGURE 4 – Arbre d'entrée pour notre algorithme d'inférence grammaticale. Certains types sont déjà traités par l'étape de transduction et les autres sont remplacés par des variables.

similarité. Lorsque, pour une hauteur donnée, nous avons plusieurs possibilités d'unification, nous appliquons un ordre de priorité qui peut être résumé par :

1. unifier les plus petits clusters,
2. unifier les clusters pour lesquels il n'y a qu'un seul choix par variable,
3. unifier avec le plus simple candidat (la complexité d'un candidat est calculée en fonction du nombre de \et de / qu'il contient),
4. choisir le premier venu pour les autres variables, avec la possibilité de choisir aléatoirement l'unification.

Il se peut que tous les mots ne soient pas représentés à un niveau donné, par conséquent il peut rester des variables dans les types après une étape de clustering. Dans ce cas, on passe à un nouveau niveau de clustering. Cette manière de procéder nous assure l'unification des variables qui apparaissent en premier lieu dans des contextes les plus similaires possibles.

Il est à noter que même avec des variables, les arbres de dérivation partiels restent des dérivations valides et représentatives d'une grammaire AB : Les deux règles d'élimination sont les seules utilisées pour créer les arbres.

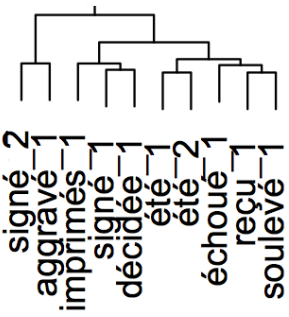


FIGURE 5 – Extrait d'un cluster de 5 phrases. Les participes passés sont regroupés d'abord par contexte puis tous ensemble dans de plus larges clusters jusqu'à n'en former plus qu'un.

4 Clustering

4.1 Extraction de vecteurs

Nous avons décidé d'assigner des vecteurs aux mots en extrayant les informations des corpus avant transductions.

Les vecteurs ont six dimensions :

1 POS-tag du mot (père),

2 information morpho-syntaxique (grand-père),

3-4 POS-tag du frère à gauche et à droite,

5-6 distance jusqu'au plus proche ancêtre commun avec le voisin de gauche et de droite.

S'il n'y a pas de voisin de droite ou de gauche (dernier ou premier mot d'une phrase), la valeur correspondant à la coordonnée de ce vecteur sera instanciée à NIL ou -5, suivant si c'est un label ou un nombre. Deux exemples de vecteurs sont donnés figure 6.

$$le_1 < \text{DET, NP-SUJ}, NIL, \text{NC}, -5, 2 >$$

$$le_2 < \text{DET, NP-MOD, VPP, ADJ}, 3, 2 >$$

FIGURE 6 – Deux vecteurs correspondant au déterminant "le".

Pour comparer les vecteurs, nous avons besoin de les transformer en vecteurs dans $\mathbb{Z}^n, n \in \mathbb{N}$. Nous avons pris le parti de transformer chaque label en vecteur où seulement une ou deux dimensions possède la valeur 1 et le reste des coordonnées a pour valeur 0. Les POS-tags et les informations syntaxiques sont transformés de cette manière. Les distances numériques restent telles quelles, comme montré figure 6. La transformation est illustrée par la table 4 avec seulement une portion des données. Il y a une "dimension" pour presque chacun des POS-tags (avec cependant quelques exceptions pour des cas que nous souhaitons voir unifiés ensemble,

comme *ET* pour les mots étrangers et *NPP* pour les noms propres) ; pour les informations morpho-syntaxiques, en plus d'une dimension pour chaque catégorie de base (*NP*, *PP*...) on fait seulement la différence entre les arguments (représentés par le *-SUJ, -OBJ, -ATS*... à la fin des labels) et les modificateurs *-MOD*.

POS-tag	**NC**	**DET**	**P**	**...**
NC	1	0	0	0...0
DET	0	1	0	0...0
P+D	0	1	1	0...0
Other	**NP**	**...**	**-ARG**	**-MOD**
NP	1	0...0		
NP-SUJ	1	0...0	1	0
NP-MOD	1	0...0	0	1

TABLE 4 – Exemple de transformation de vecteurs.

4.2 Création des clusters

Pour calculer le cluster hiérarchique nous utilisons le logiciel R (Ihaka et Gentleman, 1993), la distance métrique Manhattan et pour le clustering en lui-même la méthode de variance minimum de Ward (Ward, 1963). Pour mieux visualiser le cluster complet nous utilisons Tulip (Auber et Mary, 2007), ce qui permet de créer des graphes comme ceux de la figure 7. Les détails du graphe seront montrés dans la section suivante.

5 Evaluation

Nous avons testé notre méthode sur 754 phrase de Sequoia et 553 phrases du corpus de Paris VII. Le tableau 5 montre l'efficacité de la méthode, calculée en fonction du pourcentage de variables restant après unification.

Corpus	Sequoia (754 phrases)	Paris VII (553 phrases)
variables restantes	3	0
nombre total de variables	1.429	686
ratio	99,8%	100%

TABLE 5 – Pourcentage de variables restantes après unifications sur l'extrait de Sequoia et Paris VII.

Cependant le nombre de variables restantes sont un faible critère de succès. En effet, si les variables sont unifiées mais que les types résultants sont trop complexes, on ne peut pas dire que notre méthode soit un réel succès, même si toutes les phrases ont un arbre de dérivation valide.

Pour le corpus Sequoia, lorsque l'on compare les lexiques extraits après transduction et avec notre nouvel algorithme d'inférence grammaticale, on peut noter que 82,7% des lexiques sont identiques : cela signifie qu'il y a seulement 2967 paires mot-type différentes sur les 17110

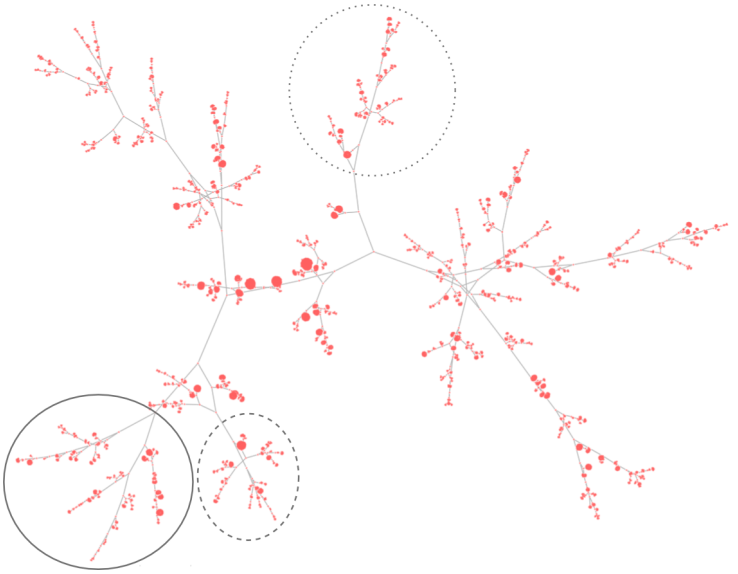


FIGURE 7 – Cluster correspondant à notre ensemble test de phrases. La partie entourée en pointillés correspond à l'endroit où il y a le plus de verbes ; celle entourée par des tirets aux déterminants et la partie simplement cerclée correspond aux adjectifs. On peut noter que les adjectifs et les déterminants sont proches les uns des autres. Cela s'explique parce qu'ils prennent généralement tous les deux un nom commun en argument et qu'ils sont présents dans des groupes nominaux.

paires du lexique. Cela correspond à 1012 entrées dans le lexique qui ont au moins une différence.

Pour les 553 phrases du corpus de Paris VII, nous comparons plus en détail les deux lexiques. Cela correspond à 2076 mots, soit 5731 paires mot-types.

Les différences entre les deux lexiques correspondent à 899 paires qui s'étalent sur 379 mots, soit 14,9% du lexique. Cela signifie que 85,1% des lexiques sont identiques. Dans ces 85,1% il faut noter cependant qu'il y a 2% de modifications mineures, telles qu'un *np* qui devient un *n* (majoritairement dans des cas tels que "Le président Merem") ou qu'une inversion entre les différents types des prépositions, pp, pp_{de} ou pp_{a} (les trois correspondent à des syntagmes prépositionnels, mais les deux derniers ajoutent comme information que la préposition utilisée est un *à* ou un *de*. Il faut noter cependant que certaines prépositions ne sont pas annotées comme ayant un *à* ou un *de* dans le corpus). Nous travaillons actuellement à faire disparaître ces modifications mineures.

Le tableau 6 trie les différences en deux catégories : d'un côté les types qui sont présents dans le lexique provenant du transducteur mais qui n'ont pas la même occurrence, et de l'autre ceux qui n'apparaissent pas dans le lexique de référence. Quelques exemples sont montrés dans le tableau 7. Le participe passé *accumulé* peut être utilisé aussi bien comme un adjectif. Le type donné par l'unification correspond à un noeud VPP utilisé comme un participe passé et non comme un

adjectif, ce qui pourtant correspond mieux au contexte, cependant la *CCG Bank* (Hockenmaier et Steedman, 2007) contient une règle spéciale qui permet une translation de $np\backslash s_p$ à $n\backslash n$; on peut donc dire que le fait de considérer les deux types comme équivalents pour l'évaluation semble être justifié.

Le type donné à *change* est une vraie erreur : à la place d'être traité comme un verbe transitif qui prendrait donc deux arguments, il est traité comme un verbe intransitif. Cette erreur vient de l'étape de clustering, où *change* est proche d'un autre verbe intransitif.

paires erronées	569	8,7%
paires équivalentes	336	6,2%
paires identiques	4 832	85,1%
paires utilisables	5 168	**91,3%**

TABLE 6 – Ratio entre les différences des lexiques, comptées en paire mot-type. On note que 91,3% du lexique est sans erreur, donc utilisable en l'état.

Mot	**Unification**	**Transduction**
accumulé	$np\backslash s_p$	$n\backslash n$
change	$np\backslash s$	$(np\backslash s)/np$

TABLE 7 – Un exemple de chaque classe de mots.

La figure 8 montre deux clusters de niveau zéro. Le gauche est un cluster d'adverbes. La variable ab_{331} sera unifiée avec np. Le cluster de droite contient uniquement des variables qui seront unifiées en une seule. Il rassemble des adjectifs et des participes passés utilisés en tant qu'adjectifs.

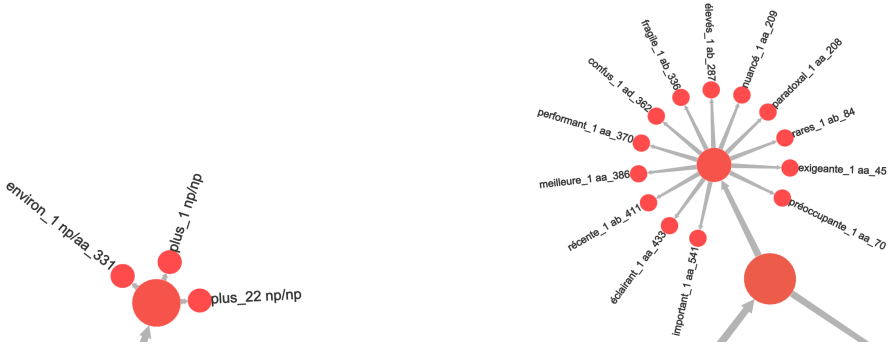


FIGURE 8 – Zoom sur deux clusters de niveau zéro.

6 Discussions

La méthode que nous utilisons est faite pour fonctionner avec des arbres de dérivation : il s'agit d'apprentissage non supervisé, mais avec des structures d'entrée contenant beaucoup d'informations dont des informations syntaxiques. Cependant, cela pourrait être étendu à n'importe

quel ensemble de phrases qui ne sont pas sous forme d'arbres avec quelques modifications. L'idée serait d'utiliser à la fois des phrases simples et d'autres phrases sous forme d'arbres de dérivation.

Le problème est d'avoir les vecteurs de mots pour les phrases qui n'ont pas d'arbres syntaxiques attachés. On pourrait alors utiliser les vecteurs d'un sous-espace car certaines informations, comme le POS-tag des mots, peuvent être facilement retrouvées avec un tagger (Moot, 2012).

Ensuite, nous pouvons effectuer une étape de clustering avec ces vecteurs partiels et ceux extraits d'arbres syntaxiques en faisant une projection sur les seconds pour diminuer le nombre de dimensions. De cette manière, les mots pourraient avoir le type le plus utilisé par le cluster de niveau zéro leur correspondant. Cela nous permettrait d'avoir une plus grande visibilité sur les mots que si nous leur donnions juste le type le plus utilisé dans un lexique de référence en fonction de leur POS-tag. Etant donné que nos vecteurs ont un grand nombre de dimensions et sont très vides, nous pourrions aussi appliquer la méthode décrite par Kailing et al. (Kailing *et al.*, 2004) pour les manipuler.

6.1 Application à de plus grands corpus

Nous souhaitons appliquer notre méthode actuelle à des ensembles plus larges, mais nous aurons alors affaire à des clusters beaucoup plus larges pour le corpus Sequoia complet (plus de 63000 mots) ou encore pour le corpus de Paris VII (environ 300000 mots). L'étape de clustering est, avec la méthode de Ward (Ward, 1963), d'une complexité $O(n^3)$, et cela commence à devenir problématique pour ces grands ensembles. Il faut cependant noter que cela constitue une amélioration par rapport aux autres algorithmes d'apprentissage, étant donné que les grammaires k-valuées ont une complexité exponentielle.

6.2 Optimisation de l'unification des types

Pour l'instant, nous utilisons le critère "premier trouvé" pour unifier les variables lorsque nous n'avons pas d'autre critère de choix. Une solution plus optimale serait de regarder toutes les variables dans leur globalité, de leur assigner une liste d'unifications possibles et d'utiliser l'algorithme de Kuhn-Munkres (Kuhn, 1955; Munkres, 1957) pour choisir la meilleure unification globale, comme par exemple celle qui donne l'instantiation des variables avec les types les plus simples.

7 Conclusion

Dans cet article, nous avons montré une nouvelle méthode pour extraire une grammaire AB par unification en utilisant le clustering pour nous guider. Une implémentation est disponible (Sandillon-Rezer, 2013).

Nous avons décidé d'utiliser un clustering hiérarchique, ce qui nous permettait d'unifier le lexique pas à pas, soit jusqu'à convergence, soit jusqu'à ce qu'un conflit bloque l'unification. Cependant, il serait intéressant de tester notre méthode avec d'autres types de clustering, comme la méthode des *k-means* ou celle appelée *Clustering By Committee* (Pantel, 2003). Cette dernière méthode

cherche le meilleur centroïde de chaque cluster qui est sensé être représentatif de chacun ; elle ne peut cependant pas être appliqué en l'état, parce que nous souhaitons faire l'unification après le clustering et que les types des centroïdes ne sont donc pas encore définis.

Les résultats que nous avons sont prometteurs surtout si on garde à l'esprit le fait que le format d'entrée est peu détaillé, ce qui nécessite donc moins d'heures aux annotateurs, mais que nous sommes proches de notre lexique de référence : nous avons 91,3% du lexique qui est similaire et 85,1% qui est identique.

Références

Abeillé, A. et Clément, L. (2003). Annotation morpho-syntaxique.

Abeillé, A., Clément, L. et Toussenel, F. (2003). Building a treebank for french. *Treebanks, Kluwer, Dordrecht*.

Adriaans, P. W. (1999). Learning shallow Context-Free languages under simple distributions.

Ajdukiewicz, K. (1935). Die syntaktische konnexität. *Stud. Philos.*, 1:1–27.

Auber, D. et Mary, P. (2007). Tulip : Better visualization through research.

Bar-Hillel, Y. (1964). *Language and information : selected essays on their theory and application*. Addison-Wesley Pub. Co.

Buszkowski, W. et Penn, G. (1990). Categorial grammars determined from linguistic data by unification. *Studia Logica*.

Candito, M. et Seddah, D. (2012). Le corpussequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical.

Costa-Florêncio, C. (2001). Consistent identification in the limit of any of the classes k-valued is np-hard. *Lecture Notes in Artificial Intelligence*.

Gold, E. (1967). Language identification in the limit. *Information and Control*, 10.

Hockenmaier, J. (2003). Data and models for statistical parsing with combinatory categorial grammar.

Hockenmaier, J. et Steedman, M. (2007). CCGbank : a corpus of CCG derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396.

Ihaka, R. et Gentleman, R. (1993). R project.

Kailing, K., Kriegel, H. et Kröger, P. (2004). Density-connected subspace clustering for high-dimensional data. *Proceedings of the Fourth SIAM International Conference on Data Mining*.

Kanazawa, M. (1998). *Learnable Classes of Categorial Grammars*. Center for the Study of Language and Information, Stanford University.

Kuhn, H. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*.

Lambek, J. (1958). The mathematics of sentence structure. *The American Mathematical Monthly*, 65.

Moot, R. (2012). Wide-coverage semantics for spatio-temporal reasoning. *Traitement Automatique des Langues 52*.

MUNKRES, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*.

PANTEL, P. (2003). Clustering by committee. *PhD thesis*.

SANDILLON-REZER, N.-F. (2013). http ://www.labri.fr/perso/nfsr.

SANDILLON-REZER, N.-F. et MOOT, R. (2011). Using tree tranducers for grammatical inference. *Proceedings of Logical Aspects of Computational Linguistics 2011*.

TRAUTWEIN, H., ADRIAANS, P. et VERVOORT, M. (2000). Towards high speed grammar induction on large text corpora.

WARD, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*.

Améliorer l'extraction et la description d'expressions polylexicales grâce aux règles transformationnelles

Aurélie JOSEPH[1,2]
(1) LDI, 99 avenue Jean-Baptise Clément F-93430 Villetaneuse
(2) ITESOFT, Parc d'Andron, le Séquoia, 30470 Aimargues
joseph.aurelie@gmail.com

RÉSUMÉ

Cet article présente une méthodologie permettant d'extraire et de décrire des locutions verbales vis-à-vis de leur comportement transformationnel. Plusieurs objectifs sont ciblés : 1) extraire automatiquement les expressions phraséologiques et en particulier les expressions figées, 2) décrire linguistiquement le comportement des phraséologismes 3) comparer les méthodes statistiques et notre approche et enfin 4) montrer l'importance de ces expressions dans un outil de classification de textes.

ABSTRACT

Enhance Multiword Expressions Extraction and Description with Transformational Rules

This paper presents a methodology to extract and describe verbal multiword expressions using their transformational behavior. Several objectives are targeted: 1) automatically extracting MWE and especially frozen expression, 2) describing linguistically their MWE behavior, 3) comparing statistical methods and our approach, and finally 4) showing the importance of MWE in a text classification tool.

MOTS-CLÉS: expressions polylexicales, expressions figées, locution verbale, extraction, transformation, classification de textes
KEYWORDS : multiword expression, verbal phrase, extraction, transformation, text classification

1 Introduction

Depuis quelques années l'extraction d'expressions phraséologiques (EP) est devenue un problème majeur dans le traitement du langage naturel. Du fait de l'universalité du phénomène à travers les langues, de l'importance des EP dans les corpus et de leur impact dans la compréhension, il devient primordial de décrire et de traiter cet objet linguistique

Notre principal but est de proposer une méthodologie permettant d'extraire ces mots polylexicaux englobant ce qui est connu sous le nom de noms composés (*pomme de terre, lune de miel*), locutions verbales (*prendre en compte, mettre fin*), idiomes (*casser sa pipe, peigner la girafe*), collocations (porter plainte)...

La plupart des méthodologies extraient généralement des expressions qui sont en réalité des éléments terminologiques ou des collocations. Peu d'études utilisent les contraintes transformationnelles qui sont des critères incontournables des EP. Nous voulons démontrer qu'une approche transformationnelle permet de retrouver ces expressions et même de donner des critères permettant de les décrire et ainsi de les catégoriser dans leur degré de figement. De plus, la majorité des approches traitent les expressions nominales (Daille, 1996 ; Watrin, 2007 ; François 2011). Dans cet article, nous nous focalisons sur les EP verbales (Kilgarriff, 2002 ; Smadja, 1993) qui sont plus difficiles à traiter.

2 Etat de l'art

Abordons brièvement les typologies liées aux phraséologismes et aux terminologies associées. Pour plus de détails, nous renvoyons à Mejri (2011), Gross (1996 ; 2012) ou Mel'čuk (2011) pour le français ou encore Abu Ssaydeh (2005) pour l'anglais. Dans notre étude nous nous concentrons sur :

- Les expressions totalement figées. Elles n'acceptent aucune variation, leur sens est souvent opaque et elles sont lexicalisées : *au fur et à mesure.*
- Les expressions semi-figées qui acceptent quelques variations. C'est ici que la plupart des séquences verbales se situent : *prendre une veste, casser sa pipe.*
- Les collocations. Ce sont des expressions qui « aiment » être ensemble (intimer l'ordre) mais dont le comportement syntaxique reste assez libre.
- Les routines langagières (*veuillez agréer mes sincères salutations*).

De nombreuses méthodes sont utilisées pour extraire ces éléments.

2.1 Les approches statistiques

Les mesures probabilistes telles que le rapport de vraisemblance ou la mesure de Dice, sont très souvent utilisées par les chercheurs pour déterminer les termes apparaissant fréquemment ensemble (Sinclair, 1991). Dias (2003) propose également une méthode sans ressource linguistique, utilisable indépendamment de la langue et sans contrainte dans le nombre de mots possibles dans la séquence. Les méthodes statistiques ont l'avantage d'être facilement implémentables, rapides et efficaces dans leur traitement mais laissent souvent de côté les expressions figées (Ramisch 2012) en faveur des

collocations. Elles nécessitent également des corpus volumineux.

2.2 Les approches hybrides

Même si certains chercheurs refusent les ressources linguistiques à cause de ses inconvénients (dépendantes de la langue étudiée, souvent longues à construire et à maintenir), les méthodologies les plus performantes et les plus utilisées, combinent statistique et des filtres linguistiques. Ces filtres peuvent être des nettoyages de mots grammaticaux (Manning et Schütze, 1999), des sélections de structures syntaxiques productives (Watrin, 2007). Quelquefois les chercheurs introduisent les transformations pour étendre leur extraction (Daille, 1996) ou pour vérifier la validité des candidats (Al Haj et Wintner, 2010 ; Abeillé et Schabes, 1989). L'outil mwetoolkit (Ramisch 2012) propose de nombreuses possibilités pour extraire ces expressions selon certains filtres.

2.3 Les approches multilingues

Très brièvement revenons sur les travaux de Villada Moirón et al. (2006) ou Archer (2006) qui proposent une extraction basée sur la comparaison de corpus parallèles. Ils partent du postulat que certaines EP ne sont pas traduisibles mot à mot. En d'autres termes, la traduction de chaque terme ne peut mener à la traduction de l'expression entière dans la langue cible.

2.4 Les approches basées sur des ressources linguistiques

Alors que des listes répertoriant les expressions figées peuvent être utilisées (Grezka et Poudat, 2012), pour reconnaître les expressions semi-figées qui sont les EP les plus fréquentes dans les corpus, la meilleure ressource est celle qui décrit les variations des composants et les transformations possibles. Le lexique-grammaire (initié par M. Gross) et les ressources du LDI (Ben-Henia Ayat, 2006 ; Cartier, 2010 ; Buvet, 2008) décrivent chaque expression de cette manière. Mais la description est très coûteuse en temps de réalisation. C'est pourquoi nous voulons améliorer cette approche en introduisant des descriptions automatiques.

Plus récemment, les méthodes basées sur le Web, ont émergé. Certaines utilisent les moteurs de recherche (Colson, 2010 ; Cartier et Joseph, 2011), d'autres utilisent Google ngrams (François, 2011) ou Wikipédia (Garcia-Fernandez et al., 2011).

3 Le corpus

Dans cette étude, nous utilisons un corpus existant dans notre entreprise composé de lettres écrites par des clients (1 533 documents, 273 669 formes). Ces courriers sont dactylographiés et concernent la relation entre un client et une entreprise télévisuelle. Chaque classe représente un sujet particulier (gestion d'abonnement, offres, annulation simple, annulation complexe, réclamation offre, réclamation financière). Le niveau d'orthographe et de grammaire du client entraîne inévitablement des erreurs qui ne sont pas corrigées. Cependant, la reconnaissance optique des caractères liée à la numérisation des courriers et les erreurs d'orthographes ne représentent que 2% des formes. Ce corpus spécialisé nous permet de nous confronter à des données réelles, problématiques et nous

permet d'appréhender la phraséologie de ce domaine.

4 Approche méthodologique

Notre approche utilise les contraintes transformationnelles liées aux EP (substitutions, insertion de modifieurs, passivation…) afin de les extraire, de les catégoriser automatiquement et de décrire leur comportement. L'extraction et la description sont réalisées à partir d'un corpus.

4.1 Architecture du système

Voici les différents processus pour extraire les phraséologismes :

- Un étiquetage morphosyntaxique permettant d'extraire une liste de candidats dans un corpus à partir de structures syntaxiques.
- La création d'un programme générant pour chaque candidat, les transformations possibles.
- La création d'un programme qui recherche les transformations dans les textes.

4.2 L'extraction par structures syntaxiques

Il a été prouvé depuis quelques années que les EP et en particulier les expressions verbales ne sont pas syntaxiquement déviantes de la langue comme avait pu le postuler Björkman (1978). Au contraire, la plupart des EP correspondent à des structures de la syntaxe libre. Les structures les plus productives (appelées moules syntaxiques) peuvent être listées. Par exemple, pour les noms : Mathieu-Colas (1988) ; pour les verbes : Gross (1982), Schmid (1991), Cartier et Joseph (2011) ; pour les adverbes : Grezka et Poudat (2012).

Dans cet article nous nous limitons aux EP verbales composées d'un seul nom car ce sont les structures les plus productives. Nous étudions donc les structures suivantes : VERBE NOM ; VERBE DETERMINANT NOM ; VERBE PREPOSITION NOM ; VERBE PREPOSITION DETERMINANT NOM. Après avoir taggué et lemmatisé notre corpus avec Treetagger, nous extrayons les EP candidats à partir de ces structures.

4.3 Transformations morphologiques, syntagmatiques et paradigmatiques

Les règles transformationnelles testées sont celles expliquées dans la littérature (Gross 1996, Lamiroy 2008…). Cependant, certaines sont supprimées car trop vagues pour être traitées automatiquement. Parmi elles, mentionnons la pronominalisation.

*Contacter votre service → le contacter ; Faire un geste → *en faire un*

Nous ne testons pas non plus l'insertion d'adverbes entre le verbe et le complément car la majorité des séquences verbales acceptent ce modifieur faisant d'elles des séquences semi-figées. Finalement trois grands types de transformations sont testés.

4.3.1 Les transformations morphologiques

Elles correspondent aux variations des composants, comme :

- La flexion nominale : le nombre du nom (*une étude, des études*)
- La nominalisation

G. Gross (2010) argue que ce critère se limite aux verbes prédicatifs (*résilier un contrat → une résiliation de contrat*). Mais selon M. Gross (1986), certaines expressions figées acceptent la nominalisation (*Mettre en scène → une mise en scène)*.

Ce critère peut toutefois permettre de catégoriser un grand nombre de prédicats. Notons également que la nominalisation peut entraîner des modifications syntagmatiques.

4.3.2 Les transformations syntagmatiques

Elles correspondent à des modifications sur l'axe syntagmatique, liées aux règles d'ordre des mots.

- Clivage : **C'est une fin que je mets à mon contrat*
- Passivation : *?Une fin est mise à mon contrat*
- Relativation : **La fin que je mets à mon contrat*

Les variations syntagmatiques peuvent également être dues à :

- La suppression ou l'insertion de déterminant : *Je mets une fin à mon contrat ; *Je fais point sur cette situation*
- L'insertion de modifieurs : *??se renvoyer la petite balle*
- L'insertion de syntagme entre le verbe et le complément : *?Faire sur la situation le point*

4.3.3 Les transformations paradigmatiques

Ces transformations concernent les substitutions des composants de la séquence avec un composant de même nature (un verbe avec un verbe, un nom avec un nom…). Contrairement aux méthodes statistiques, nous évaluons les substitutions possibles ayant la même structure syntaxique. Par exemple, pour la séquence *mettre fin,* 13 verbes de notre corpus sont substituables avec *mettre* (*avoir en fin, arrêter à la fin, attendre la fin*…). Cependant, en ajoutant une contrainte structurelle (*mettre fin* correspond à la structure VER NOM), seulement 4 sont conservées (*prendre fin*, *donner fin*, *arrêter fin* et *prévenir fin)*.

De plus, nous ne gardons que les substitutions possédant une catégorie grammaticale du contexte droit identique à celles possibles dans la séquence source. *Mettre fin* possède 3 contextes possibles : a) une préposition : mettre fin **à** mon contrat, b) une ponctuation (point, virgule…) : j'y mets fin**.**, c) un adverbe : je mets fin **immédiatement** à mon contrat. Parmi les substitutions précédemment sélectionnées, seules 2 séquences (*prendre fin* et *donner fin*) ont un contexte en commun avec *mettre fin* (ici une préposition : *prendre fin **en** septembre, donner fin **à** mon abonnement*). Ici, nous remarquons que la préposition à elle seule ne suffit pas pour montrer que nous avons deux contextes identiques (l'un complément d'objet indirect, l'autre complément circonstanciel de temps). Il faudrait améliorer l'analyse. Mais nous réduisons déjà un certain nombre de possibilités substitutionnelles.

Enfin, nous ne comptons pas le nombre d'occurrences de chaque substitution dans le corpus, mais le nombre de substitutions différentes. En d'autres termes peu importe le nombre de fois où *prendre fin* ou *donner fin* apparaissent dans le corpus seule compte le nombre de formes différentes substituables (ici 2).

4.3.4 Implémentation

Les transformations sont implémentées en utilisant de simples expressions régulières. Nous n'utilisons pas actuellement de ressources externes comme un parseur. Toutefois, nous sommes conscients de l'utilité de ces outils pour améliorer les traitements et résultats (Wehrli et al. 2010).

4.4 Seuil de fréquence, de la "règle de trois" à la "règle de deux"

Nous appelons règle de trois la méthode utilisée pour déterminer que les séquences sont utilisées assez fréquemment et possèdent une dispersion assez significatives pour être appelées collocations (Dubreil et Daille 2005). Pour cela, 3 règles doivent être vérifiées

- « - la cooccurrence de deux termes apparaissant au moins trois fois dans le corpus ;
- la cooccurrence de deux termes issus de trois articles différents;
- la cooccurrence de deux termes employés par trois auteurs différents. » (Dubreil et Daille, 2005)

Dans notre cas, ces trois critères peuvent être réduits à seulement deux. Chaque courrier est écrit par un client unique, donc « trois auteurs différents » et redondant avec « trois articles différents ». De plus, notre corpus n'étant pas très volumineux, nous réduisons l'apparition des termes au moins trois fois dans le corpus par seulement deux fois. De plus, n'oublions pas que nous voulons extraire également des séquences figées qui dans un corpus réduit ne sont pas très fréquentes. Ce seuil mis à « deux » est un bon compromis entre la non significativité d'une apparition unique et un seuil trop haut.

4.5 Score de figement

Le score de fixité permet de décider si une séquence candidate est une expression phraséologique. Pour comprendre ce score précisons son calcul. Tout d'abord, nous calculons un score pour chaque type de transformation (syntagmatique, morphologique et paradigmatique). Le calcul du score pour les transformations syntagmatiques et morphologiques est identique. C'est un simple ratio entre le nombre d'occurrences de la séquence candidate et la somme des différents tests et de la séquence candidate.

$$F_{synt}{}^{Si} = \frac{S_n}{(S_n + \sum T_n)}$$

Le calcul du score du figement paradigmatique est un ensemble d'heuristiques. Il dépend à la fois du nombre de déterminants substituables de la structure syntaxique et des substitutions verbales et nominales. Une différence entre les « upward collocations » et les « downward collocations » est également faite (Sinclair 1991). Les « upward collocations » sont dans notre cas des substitutions verbales, tandis que les « downward

collocations » sont des substitutions nominales. Disons brièvement que les « downward collocations » peuvent être très productives surtout avec des auxiliaires. Par exemple, dans notre corpus, *avoir* accepte 99 noms différents. Mais *fin*, nous l'avons vu, n'accepte que 2 substitutions dans une structure VER NOM. Donc, nous mettons en valeur les « upward collocations » c'est-à-dire les substitutions verbales possibles à partir d'un nom. C'est toutefois un choix risqué lorsque l'on sait que les collocations se réalisent majoritairement à partir du prédicat.

Enfin, nous fixons plusieurs seuils permettant de déduire qu'un candidat est une expression phraséologique : le score de figement morphologique (F-M) doit être supérieur à 0.8 ; le score de figement syntagmatique (F-S) doit être supérieur à 0.7 ; le score de figement paradigmatique (F-P) doit être supérieur à 0.6. Ces choix de seuils sont pour le moment pris de manière subjective, respectant tout de même une certaine probabilité dans les différentes transformations.

En prenant trois scores différents nous pouvons contrôler les seuils. L'application d'une moyenne entre les scores ou du calcul d'un score plus général entraîne plus de bruit.

4.6 Constitution d'une base de référence

Afin de comparer notre approche avec une base de référence, nous devons constituer cette base. Actuellement elle est réalisée grâce à différentes sources répertoriant des expressions figées (Lexique-Grammaire[1], expressio[2], DEL[3], Wiktionary[4]). Nous faisons remarquer immédiatement que le terme expression figée est pris de manière assez large selon les ressources. Des verbes supports peuvent même apparaître (*mettre fin*). Toutefois ils sont considérés comme des combinaisons avec une forte attraction et devenir des locutions comme en témoignent certains dictionnaires (notamment le TLFI). Nous les considérons de manière assez naïve sans remettre en considération leur présence. Finalement, 110 séquences évaluées comme étant figées par ces ressources ont été trouvées parmi nos candidats.

5 Résultats

L'extraction à partir de structures syntaxiques abouti à 5 148 candidats représentant 15 794 formes différentes (incluant les transformations morphologiques et syntagmatiques). 1 133 séquences peuvent prétendre au titre d'expressions phraséologiques après l'application de la règle de 2. Parmi elles 302 séquences sont assez figées pour être considérées comme des EP par notre approche, c'est-à-dire en appliquant les scores de figement.

[1] Revu et corrigé par Tolone 2011

[2] Georges Planelles 2011

[3] *Dictionnaire des expressions et des locutions* Rey et Chatreau 2006

[4] http://fr.wiktionary.org/wiki/Catégorie:Locutions verbales en français

5.1 Distribution des transformations

Les transformations représentent 44% des formes. Certaines transformations sont beaucoup plus utilisées que d'autres. Certaines sont même pratiquement inutiles. Par exemple, le clivage n'est utilisé que 3 fois dans notre corpus. Selon Riegel et al. (1994), le clivage est plus utilisé dans le langage parlé qu'écrit. Nous le vérifions ici également.

Règles transformationnelles	**Occurrences**
Insertion syntagme	8,09%
Insertion déterminant	2,42%
Inversion	4,66%
Clivage	0,06%
Relativation	3,68%
Passivation	9,05%
Nominalisation	5,75%
Insertion Modifieurs	3,72%
Flexion	4,33%
Suppression déterminant	0,95%

TABLE 1 – Distribution des transformations

La nominalisation est une transformation productive et intéressante. Les verbes impliqués sont des prédicats de premier ou de second-ordre.

Résilier = résiliation ; restituer = restitution ; rembourser = remboursement
Demander (résiliation) = demande ; attendre (confirmation) = attente

La passivation quant à elle, est une des transformations les plus utilisés. Autant les prédicats verbaux que les prédicats nominaux sont touchés par cette transformation.

Prélever la somme = la somme prélevée
Effectuer un prélèvement = un prélèvement est effectué

5.2 Extraction de phraséologismes

Avant de comparer les résultats avec des méthodes statistiques, regardons combien nous retrouvons de phraséologismes répertoriés par les ressources. 110 EP répertoriées par nos ressources externes sont présentes dans notre corpus. En appliquant la règle de 2, 57 sont conservées. Notre approche en retrouve 47 (soit 80%).

Séquences	F-S	F-P	F-M
mettre fin	1	0,97	1
prendre acte	1	0,96	1
mettre un terme	1	0,97	1
prendre fin	0,87	0,92	1
tenir compte	1	0,88	1
faire l'objet	1	1	1
tomber en panne	1	0,88	1
rentrer dans l'ordre	1	0,81	1
faire le point	1	1	1
faire foi	1	1	1
renvoyer la balle	1	1	1
porter plainte	1	1	1
mener en bateau	1	1	1
couronner le tout	1	1	1

TABLE 2 – Phraséologismes extraits avec la méthode linguistique grâce aux différents scores de figement syntagmatique (F-S) paradigmatique (F-P) et morphologique (F-M).

Ces résultats montrent un échantillon des expressions classifiées comme étant figées à la fois par les ressources externes et l'approche linguistique. Pour chaque séquence, les scores de figement sont spécifiés. Nous pouvons voir que certaines ne sont pas totalement figées. Toutes les séquences extraites n'ont pas le même degré de figement. Nous avons : a) des séquences figées avec un sens opaque : *Mener en bateau, renvoyer la balle ;* b) des collocations ou des verbes supports: *mettre un terme, mettre fin, porter plainte.* Par conséquent, la plupart des séquences non catégorisées comme figées sont à juste titre, des collocations libres ou des prédicats. Par exemple, *souscrire un abonnement* est un prédicat approprié pourtant catégorisé comme une séquence figée. Notre approche confirme bien qu'au vu du comportement transformationnel, nous n'avons en aucun cas une séquence figée (F-S : 0.29, F-P : 0.36).

Séquences	F-S	F-P	F-M
souscrire un abonnement	0,29	0,36	1
avoir droit	0,55	1	1
faire appel	0,63	1	1
souscrire un contrat	0,14	0,5	1
faire un plaisir	0,67	1	1
donner l'ordre	0,33	1	1
avoir le minimum	0,5	0,5	1

TABLE 3 – Phraséologismes non extraits par la méthode linguistique

5.3 Comparaison avec les approches statistiques

Nous comparons ces résultats avec les catégorisations proposées par les méthodes statistiques. Afin de ne pas valoriser une mesure plus qu'une autre, nous prenons en considération plusieurs algorithmes de l'état de l'art : l'information mutuelle spécifique (PMI) ; le rapport de vraisemblance (LL), le chi carré (X^2) et la mesure Dice. Pour chaque mesure statistique, le rang du candidat est trié par son score. Son rang final est déterminé par la médiane de tous les rangs.

Séquences	Rang	PMI Rang	LL Rang	X^2 Rang	Dice Rang
mener en bateau	1	1	125	1	1
calculer au prorata	1	1	125	1	1
adhérer à la convention	3	2	135	3	3
remercier par avance	4	298	1	5	2
raccrocher au nez	7	11	40	2	1
couronner le tout	7	3	144	6	7

TABLE 4 – Rang des séquences candidates selon les différentes mesures statistiques

Toutes les mesures donnent plus ou moins les mêmes résultats, même si le LL dévie étrangement des autres mesures. Mais, mise à part l'attraction entre les termes, nous ne connaissons pas le comportement indiquant si nous avons tous les éléments propres aux phraséologismes. Nous remarquons également que la structure VER PRP (DET) NOM est la plus fréquente dans l'extraction. Afin de comparer ces résultats aux nôtres, nous décidons de montrer combien de candidats doivent être extraits pour trouver un maximum d'EP répertoriées.

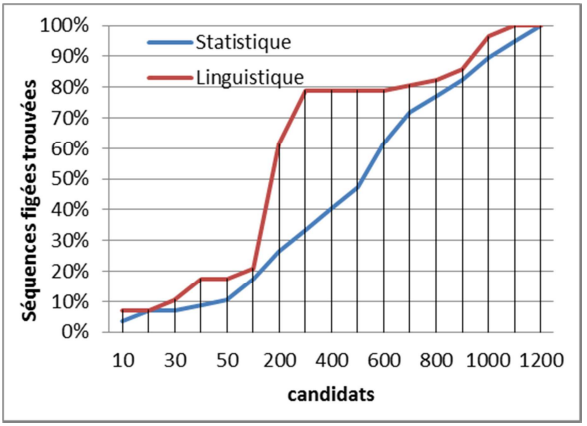


Figure 1 – comparaisons entre les approches linguistiques et statistiques pour extraire des phraséologismes parmi les candidats

Notre méthode extrait 80% des phraséologismes dès 300 séquences extraites tandis que

les statistiques ont besoin de 900 séquences candidates pour atteindre le même score. L'approche linguistique est donc plus précise et trouve les séquences figées plus rapidement que les statistiques dans un corpus non volumineux et spécialisé. Cependant plus de séquences ont besoin d'être annotées et cela de façon précise, afin de mieux comparer ces résultats et pouvoir faire un réel calcul rappel/précision.

6 Les expressions phraséologiques dans un outil de classification de textes

L'extraction des EP n'a pas d'intérêt si elle n'est pas intégrée dans une application fonctionnelle. Nous connaissons son importance dans le domaine de la traduction. Nous le supposons alors dans d'autres outils mais souvent sans preuve. C'est pourquoi, nous voulons connaître l'impact des EP dans la classification de textes. Nous avons alors procédé à une étude expérimentale. Pour effectuer cette tâche nous utilisons un classifieur propriétaire. Il utilise le corpus présenté précédemment. Celui-ci contient 6 classes. 2/3 des données sont utilisées par classe pour l'apprentissage et 1/3 pour le test. Après une phase d'apprentissage où les mots sont pondérés selon leur fréquence d'apparition, le classifieur utilise les K-plus proches voisins et une distance cosinus pour déterminer la classe la plus appropriée. Notre outil utilise également des conditions de rejets permettant de minimiser les confusions. Néanmoins cela implique un rappel plus faible. Nous voulons prouver que les phraséologismes aident à la classification. Nous voulons démontrer également que les séquences les plus figées ne sont pas les plus pertinentes pour cette tâche car elles correspondraient à des expressions linguistiques générales non spécifiques à une classe alors que les moins figées seraient plus proches de la terminologie et donc du sens de la classe.

Test	**Rappel**	**Précision**
Baseline	36%	82%
EP figée	44,9%	85.3%
EP moins figée	47,4%	87.7%
Toutes les EP	50,8%	87.6%

Table 5 – Rappel et précision dans la classification

Dans le tableau ci-dessus, l'hypothèse de départ semble être validée. Nous améliorons les résultats, que cela soit au niveau du rappel qu'au niveau de la précision. Cependant, les séquences les moins figées améliorent de 4% le rappel donné par les EP figées en augmentant la précision de 2,5% supplémentaires. Nous pouvons interpréter l'amélioration de la classification même avec des séquences complètement figées par le fait qu'elles ne sont pas seulement des séquences linguistiques générales (*accuser réception, faire part…*). Elles peuvent être des phraséologismes liées au sens de la classe (*mettre fin, tomber en panne*). Finalement les EP moins figées sont plus appropriées dans la classification car elles ressemblent à des collocations terminologiques (*renouveler un abonnement, résilier une option*). Notons toutefois que combinaison des deux listes améliore les résultats pour atteindre les 50% de classification. Par conséquent les premiers résultats laissent penser que les phraséologismes jouent un rôle dans

l'appréhension du sens et donc dans la classification de textes. Mais ceci nécessite une analyse plus approfondie.

7 Conclusions et perspectives

En résumé, nous proposons une méthode, appliquée aux expressions phraséologiques verbales, basée sur des critères linguistiques et en particulier sur leur comportement transformationnel. Ceci est effectué dans le but de les repérer et de les décrire automatiquement. Nous ne remettons pas en question l'intérêt des méthodes statistiques mais nous prouvons qu'elles ne sont pas assez précises et oublient souvent les expressions les plus figées notamment dans un corpus peu volumineux. Nous avons implémenté un système pour décrire semi automatiquement les variations des EP dans le but d'enrichir automatiquement une ressource composée d'expressions semi-figées. De cette manière, nous pouvons trouver de nouvelles entrées et ne sommes pas limités à une ressource finie.

Il serait prétentieux, à l'heure actuelle, de prétendre pouvoir catégoriser les EP sans aucun doute. D'une part, notre corpus est spécifique à un domaine et notre système doit être éprouvé avec un corpus plus générique et plus volumineux. D'autre part, notre base de validation mériterait une attention particulière pour séparer les réels phraséologismes des séquences libres. De plus, nous nous focalisons sur certaines structures syntaxiques, certes les plus productives, mais qui doivent être étendues à d'autres. Enfin, un réel critère sémantique est absent de notre étude. Les séquences pouvant avoir selon leur emploi, un sens littéral ou opaque ne sont pas distinguées, (par exemple r*envoyer la balle → se jeter les responsabilités les uns sur les autres*[5] ou r*envoyer une balle à quelqu'un* (dans le sport)). Pour améliorer la richesse du corpus et par conséquent l'identification des EP et de leur description nous intègrerons un module permettant d'utiliser le Web comme un corpus. Nous voulons utiliser les moteurs de recherche pour savoir si une transformation liée à une séquence existe (Joseph, 2012). Avec ce corpus nous pouvons extraire et faire une première description qui sera améliorée par l'utilisation du Web.

Enfin, dans cet article nous voulons prouver que les EP sont utiles pour la classification de textes. Les résultats préliminaires qui nécessitent plus de tests sont toutefois encourageants. Ils montrent que les séquences les plus figées sont moins significatives que celles possédant plus de variations mais permettent toutefois d'améliorer les résultats.

Références

ABEILLE, A. et SCHABES, Y. (1989). Parsing idioms in lexicalized tags. *In Actes de EACL (European Chapter of the Association for Computational Linguistics*, Manchester.

ABU-SSAYDEH, A.-F. (2005). Variation in multi-word units : the absent dimension. *Studia Anglica Posnaniensia : international review of English Studies*.

[5] Définition trouvée dans linternaute.com

AL-HAJ, H. et WINTNER, S. (2010). Identifying multi-words expressions by leveraging morphological and syntactic idiosyncrasy. *In Actes de COLING 2010 (Computational Linguistics)*, Beijing.

ARCHER, V. (2006). Acquisition semi-automatique de collocations à partir de corpus monolingues et multilingues comparables. *In Actes de TALN 2006 (RECITAL)*, Leuven.

BEN-HENIA AYAT, I. (2006). *Degrés de figement et double structuration des séquences verbales figées*. Thèse de doctorat, Paris 13, Villetaneuse.

BJÖRKMAN, S. (1978). *Le type avoir besoin. Étude sur la coalescence verbo-nominale en français*. Thèse de doctorat, Uppsala : Acta Universitatis Upsaliensis.

BUVET, P.-A. (2008). Quelle description lexicographique du figement pour le TAL ? le cas des adjectifs prédicatifs à forme complexe. In L*es séquences figées : entre langue et discours*, pages 43–54.

CARTIER, E. (2008). Repérage automatique des expressions figées : état des lieux, perspectives. In *les séquences figées: entre langue et discours,* pages 55–70.

CARTIER, E. et JOSEPH, A. (2011). Repérage automatique des séquences figées pour la classification des documents. *In La notion d'unité en sciences du langage*, Villetaneuse.

COLSON, J.-P. (2010). Automatic extraction of collocations : a new Web-based method. *In JADT 2010 (journées internationales d'analyse statistique des données textuelles)*, Sapienza.

DAILLE, B. (1996). Study and implementation of combined techniques for automatic extraction of terminology. *In.(Klavans et Resnik 1996)*, pages 49-66

DIAS, G. (2003). Multiword unit hybrid extraction. *In MWE (Workshop on multiword expressions)*, Sapporo.

DUBREIL, E. et DAILLE, B. (2005). Analyse sémantico-discursive des collocations lexicales en corpus spécialisé : la base « connaissance-s ». *In Actes de LTT 2005 (Lexicologie, Terminologie, Traduction)*, Bruxelles.

FRANÇOIS, T. (2011). *Les apports du traitement automatique du langage à la lisibilité du français langue étrangère*. Thèse de doctorat, Université Catholique de Louvain, Belgique.

GARCI-FERNANDEZ, A., LIGOZAT, A.-L., DINARELLI, M. et BERNHARD, D. (2011). Méthode pour l'archéologie linguistique : datation par combinaison d'indices temporels. *In TALN 2011 (Traitement automatique des langues naturelles)*, Montpellier.

GREZKA, A. et POUDAT, C. (2012). Building a database of french frozen adverbial phrases. *In LREC* 2012 *(Conference on Language Resources and Evaluation)*.

GROSS, G. (1996). *Les expressions figées en français noms composés et autres locutions*. Ophrys édition.

GROSS, G. (2010). Les verbes supports et l'actualisation des prédicats nominaux. *In Supports et prédicats non verbaux dans les langues du monde*, Cellule de Recherche en linguistique. Paris.

GROSS, G. (2012). *Manuel d'analyse linguistique*. Sens et Structure. Presses Universitaires du Septentrion.

GROSS, M. (1982). Une classification des phrases « figées » du français. pages 151–185.

GROSS, M. (1986). Les nominalisations d'expressions figées. *Langue française*, 69, 64–84.

JOSEPH, A. (2012). Pour un étiquetage automatique des séquences verbales figées : état de l'art et approche transformationnelle. *In RECITAL 2012 (Rencontre des étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues)*, Grenoble.

KILGARRIFF, A. (2002). Sketching words. *In (Corréard 2002)*, pages 125–137.

MANNING, C. et SCHÜTZE, H. (1999). Collocations. *In Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, Massachusetts, pages 141–177.

MATHIEU-COLAS, M. (1988). *Typologie des noms composés*. Technique 7, Paris 13, Paris.

MEJRI, S. (2011). Les dictionnaires électroniques sémantico-syntaxiques. *In (*Cardoso, Mejri, Mota), pages 159-187.

MEL'CUK, I. (2011). Tout ce que nous voulions savoir sur les phrasèmes, mais… *In Cahiers de lexicologie, revue internationale de lexicologie et de lexicographie*.

PLANELLES, G. (2012). *Les 1001 expressions préférées des français*. Editions de l'Opportu.

RAMISCH, C, (2012). Une plate-forme générique et ouverte pour l'acquisition des expressions polylexicales. *In RECITAL 2012 (Rencontre des étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues)*, Grenoble.

REY, A. et CHATREAU, S. (2006). *Dictionnaire d'expressions et locutions*. Le Robert. Paris.

RIEGEL, M., PELLAT, J.-C. et RIOUL, R. (1994). *Grammaire méthodique du français*. Quadrige Manuels. Paris, PUF édition.

SINCLAIR, J. (1991). *Corpus, concordance, collocation*. Oxford, oxford university press édition.

TOLONE, E. (2011). *Analyse syntaxique à l'aide des tables du Lexique-Grammaire du français*. Thèse de doctorat.

VILLADA MOIRON, B. et TIEDEMANN, J. (2006). Identifying idiomatic expressions using automatic word-alignment. *In MWE 2006 (Workshop on Multiword Expressions)*, Italy.

WATRIN, P. (2007). Collocations et traitement automatique des langues. *In Actes de Lexis and Grammar*, pages 1530–1536, Bonifacio (France).

WEHRLI, E., SERETAN, V. et NERIMA, L. (2010). Sentence analysis and collocation identication. *In MWE 2010 (Workshop on Multiword Expressions),* Pékin.

Construction de corpus multilingues : état de l'art

Manuela Yapomo [1, 2]
(1) LiLPa (Linguistique, Langues, Parole), EA 1339
(2) ICube - Laboratoire des sciences de l'Ingénieur, de l'Informatique et de l'Imagerie, UMR 7357
Université de Strasbourg
yapomodomkem@etu.unistra.fr

RÉSUMÉ

Les corpus multilingues sont extensivement exploités dans plusieurs branches du traitement automatique des langues. Cet article présente une vue d'ensemble des travaux en construction automatique de ces corpus. Nous traitons ce sujet en donnant premièrement un aperçu de différentes perceptions de la comparabilité. Nous examinons ensuite les principales approches de calcul de similarité, de construction et d'évaluation développées dans le domaine. Nous observons que Le calcul de la similarité textuelle se fait généralement sur la base de statistiques de corpus, de la structure de ressources ontologiques ou de la combinaison de ces deux approches. Dans un cadre multilingue avec l'utilisation d'un dictionnaire multilingue ou d'un traducteur automatique, de nombreux problèmes apparaissent. L'exploitation d'une ressource ontologique multilingue semble être une solution. En classification, la problématique de l'ajout de documents à la base initiale sans affecter la qualité des clusters demeure ouverte.

ABSTRACT

Multilingual document clustering : state of the art

Multilingual corpora are extensively exploited in several branches of natural language processing. This paper presents an overview of works in the automatic construction of such corpora. We address this topic by first providing an overview of different perceptions of comparability. We then examine the main approaches to similarity computation, construction and evaluation developed in the field. We notice that the measurement of the textual similarity is usually based on corpus statistics or the structure of ontological resources or on a combination of these two approaches. In a multilingual framework, with the use of a multilingual dictionary or a machine translator, many problems arise. The exploitation of a multilingual ontological ressource seems to be a worthy option. In clustering, the problem of adding documents to the initial base without affecting the quality of clusters remains open.

MOTS-CLÉS : corpus multilingues, comparabilité, similarité textuelle translingue, classification.

KEYWORDS: multilingual corpora, comparability, crosslingual textual similarity, classification.

1 Introduction

La tendance actuelle en Traitement Automatique des Langues (TAL) est au développement de méthodes translingues pour la conception et l'amélioration d'outils multilingues. De telles applications dépendent de la disponibilité de corpus multilingues en quantités considérables.

Ces corpus sont principalement de deux sortes : parallèles et comparables. Un corpus est dit parallèle s'il est constitué de textes sources et leurs traductions (McEnery et Xiao, 2007). Les corpus comparables quant à eux regroupent des documents ayant des caractéristiques communes. La difficulté qu'est l'acquisition de corpus exclusivement parallèles et l'importance des corpus comparables (démontrée empiriquement) ont favorisé le développement de méthodes de collecte de corpus comparables à grande échelle. Cependant, peu de travaux sur les standards de comparabilité de telles données ont été menés, les recherches se focalisant sur leur exploitation. Il y a donc une nécessité de développer des méthodes performantes fondées sur une définition précise de la comparabilité pour leur collecte (Su et Babych, 2012).

Cet article qui a pour objectif de présenter l'état de la recherche en acquisition et structuration de textes multilingues s'articule en 4 parties principales. Les corpus multilingues sont présentés en section 2. Nous abordons principalement la notion de comparabilité et les applications de ces corpus. La section 3 traite de leur construction en abordant les sources de collecte, les méthodes de calcul de la similarité et de construction de ces corpus. Nous examinons ensuite en section 4 les différentes approches intrinsèques et extrinsèques d'évaluation des données résultantes. Ces éléments nous permettrons de mieux identifier les limites du domaine et déterminer notre apport à la fois théorique et pratique en section 5. Enfin, nous concluons ce travail en section 6.

2 Corpus multilingues

2.1 La comparabilité en corpus multilingues

La capacité des corpus multilingues à améliorer la performance des systèmes qui y ont recours serait fortement liée à leur degré de comparabilité.

2.1.1 Définition de la comparabilité de documents multilingues

Plusieurs travaux mentionnent le besoin d'une définition de la comparabilité et formulent leur compréhension de celle-ci. Su et Babych (2012) mesurent la comparabilité de textes à leur potentiel d'extraction de segments parallèles et d'amélioration de la performance de systèmes de traduction automatique. Li *et al.* (2011) quant à eux considèrent deux textes ou corpus comme comparables s'ils ont une partie non négligeable de vocabulaire en commun, la principale application étant l'extraction de lexiques bilingues. Leturia *et al.* (2009) soutiennent que la définition de la comparabilité ne peut être dissociée de l'application ciblée et du type de corpus souhaité. La conception de la similarité varierait donc d'un objectif à l'autre. Elle serait également influencée par le type de corpus qui peut être général ou de spécialité. Nous pensons qu'il en est de même pour la source de collecte de documents. Les critères de comparabilité de documents obtenus de Wikipédia (Paramita *et al.*, 2012) par exemple peuvent différer de ceux de documents venant d'un domaine d'articles de presse. Cette mesure de comparabilité est définie par les critères qui la composent.

2.1.2 Choix et association de critères de comparabilité

Dans le cadre d'une application particulière, les paramètres de comparabilité n'ont pas la même préséance. Certains travaux ne prennent en compte que des critères linguistiques, d'autres des critères purement extralinguistiques et d'autres encore font usage de paramètres des deux types.

– En se focalisant principalement sur le contenu, Steinberger *et al.* (2002) mesurent la similarité de documents en comparant les représentations des contenus de documents obtenues au moyen des descripteurs d'un thesaurus. Pour l'extraction de terminologies, Goeuriot *et al.* (2009) ajoutent au thème et au domaine, le type de discours comme critère d'homogénéité. Le type de discours (science ou science populaire) est identifié à travers la structure et les aspects modaux et lexicaux des textes. Pour la même application, Leturia *et al.* (2009) se basent sur la similarité de domaine.
– Pour ce qui est de l'exploitation de critères extralinguistiques, Resnik (1999) identifie des documents parallèles sur le Web à l'aide de la structure de leurs pages. Utsuro *et al.* (2002) quant à eux déterminent la comparabilité d'articles de presse en fonction de leurs dates de publications.
– La majorité des travaux combine à la fois des critères des deux types. Ainsi, les travaux de Baradaran Hashemi *et al.* (2010) se basent sur le sujet et la date de publication pour l'obtention de documents comparables pour la traduction de requêtes en recherche d'information interlingue (RII). Aker *et al.* (2012) exploitent d'une part le sujet et d'autre part, les entités nommées et dates de publication pour une application de traduction automatique. Ils mettent l'accent sur l'importance des métadonnées dans cet exercice.

Nous observons que la tendance, indépendamment de l'application, est de prendre en compte comme critères de similarité soit le thème seul qui peut être modélisé de diverses manières soit le thème accompagné d'un ou de plusieurs autres paramètres. L'utilisation exclusive de critères extralinguistiques fournit des résultats moins bons. Pour rendre compte de la comparabilité, plusieurs échelles de comparabilité ont été développées dans la littérature. Nous établissons dans le tableau 1 une correspondance entre les différents niveaux de ces échelles.

Seuls Skadiņa *et al.* (2010) traitent de la collecte de corpus multilingues avec la mention de textes parallèles. Les autres travaux se limitent aux corpus comparables. Les échelles les plus et moins granulaires sont respectivement celles de Braschler et Schäuble (1998) à 5 niveaux et Bekavac *et al.* (2004) à 2 niveaux. Excepté cette dernière échelle, aucune autre ne considère les critères extralinguistiques comme seules composantes de la comparabilité même la plus légère. Nous remarquons qu'il n'y a pas de consensus quant aux différents degrés de comparabilité en corpus multilingues, sachant qu'à l'exception de Skadiņa *et al.* (2010), ces travaux se basent uniquement sur des articles de presse. Ces échelles de comparabilité étant établies pour le jugement humain, il se pose également le problème de leur adaption à la similarité automatique.

Le degré de comparabilité de documents multilingues jouerait un rôle crucial dans leurs applications.

2.2 Applications des corpus multilingues

L'importance des corpus en général et notamment des corpus multilingues s'observe dans plusieurs domaines du TAL. De nombreux travaux portent sur l'extraction de segments parallèles

<table>
<tr><th></th><th>Bekavac et al. (2004)</th><th>Skadiņa et al. (2010b)</th><th>Braschler & Schäuble (1998)</th><th>Pouliquen et al. (2004)</th></tr>
<tr><td rowspan="4">Critères linguistiques & extra-linguistiques</td><td></td><td>(1) parallélisme</td><td></td><td></td></tr>
<tr><td rowspan="3">(1) forte comparabilité</td><td rowspan="2">(2) forte comparabilité</td><td>(1) histoire identique</td><td>(1) article identique</td></tr>
<tr><td>(2) histoire liée</td><td>(2) article lié</td></tr>
<tr><td rowspan="2">(3) faible comparabilité</td><td>(3) aspects communs</td><td rowspan="2">(3) article vaguement lié</td></tr>
<tr><td>(4) terminologie commune</td></tr>
<tr><td>Critères extra-linguistiques (uniquement)</td><td>(2) faible comparabilité</td><td>(4) aucune comparabilité</td><td>(5) sans lien</td><td>(4) sans lien</td></tr>
</table>

TABLE 1 – Niveaux de comparabilité en corpus multilingues

à partir de corpus multilingues. Concernant l'extraction de terminologies ou de lexiques multilingues des textes, l'approche générale (Fung et Yee, 1998) consiste en la représentation des mots des textes en vecteurs de contextes. La traduction d'un mot source dans la langue cible est identifiée par le repérage de vecteurs similaires ou équivalents dans les données cibles. Le projet TTC [1] (Blancafort *et al.*, 2010) a abouti à la création d'un outil de génération automatique de terminologies bilingues à partir de corpus comparables dans plusieurs langues pour la traduction automatique. Pour cette même application, Bin *et al.* (2010) agrandissent un corpus parallèle anglais-chinois avec des phrases parallèles extraites de corpus comparables de brevets d'invention. Des résultats encourageants sont obtenus par le système de traduction automatique entraîné et testé avec le corpus parallèle obtenu. Ion (2012) propose un outil d'extraction de segments parallèles des corpus comparables pour enrichir des modèles de traduction statistique.

Réaliser ces applications nécessite d'avoir une quantité considérable de documents multilingues et donc des méthodes performantes pour leur collecte automatique.

3 Construction automatique de corpus multilingues

La procédure de construction de corpus multilingues présente 3 aspects principaux.
– les sources à partir desquelles les documents sont initialement extraits (section 3.1).
– la mesure de similarité permettant d'évaluer la comparabilité entre deux textes ou entre des textes et des groupes de textes (section 3.2).
– et enfin la méthode de construction de corpus elle-même pouvant prendre la forme de crawling, de RII ou de classification (section 3.3).

1. TTC : *Terminology Extraction, Translation Tools and Comparable Corpora*. `http://ttc.syllabs.com/`

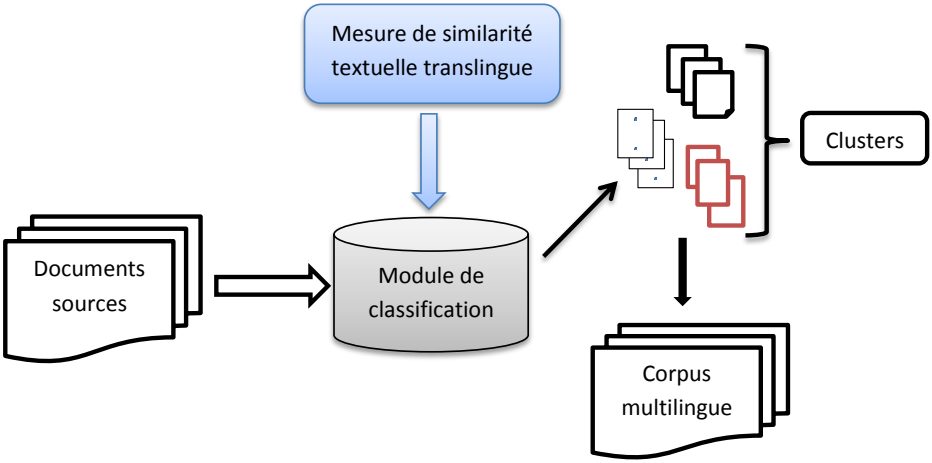


FIGURE 1 – Architecture générale d'un système de classification de documents

La méthode de construction se sert de la mesure de similarité développée pour obtenir des documents hautement comparables et/ou parallèles de la source. Cette procédure développée sous la présente section est illustrée par la figure 1. Dans cette article, nous nous orientons vers la méthode de classification, également illustrée dans la figure 1, pour la construction d'un corpus multilingue.

3.1 Sources d'acquisition

3.1.1 Collections existantes

Les recherches antérieures en compilation de corpus multilingues ont été principalement faites sur la base de corpus de recherche existants. Dans leur objectif de produire un corpus comparable, Bekavac *et al.* (2004) exploitent des sous-corpus monolingues d'un corpus de référence bulgare et croate dont ils alignent les documents comparables. Talvensaari *et al.* (2007) utilisent deux collections de documents monolingues, pour créer un corpus comparable suédois-anglais. Ces données ont l'avantage d'être prêtes pour l'utilisation puisqu'il n'est pas nécessaire de développer des méthodes supplémentaires pour la collecte comme c'est le cas avec le Web. Cependant, la variété des données reste un problème auquel le Web apporte un début de solution.

3.1.2 Le Web

Le Web est une source de plus en plus sollicitée pour la collecte de corpus en tous genres. De nombreux travaux en acquisition de corpus multilingues sur le Web se focalisent sur les articles de presse. Aker *et al.* (2012) tirent profit de la représentation des documents en clusters et de la disponibilité des titres à travers les flux RSS qu'offre *Google News* pour extraire des articles

comparables pour les paires de langues anglais-allemand et anglais-grec. Wikipédia aussi est utilisé pour cette application grâce à sa structure ou à ses liens inter-langues qui mettent en relation les articles de différentes langues sur des thèmes identiques ou liés. Ion *et al.* (2010) compilent un corpus comparable roumain-anglais par l'identification d'articles ayant les mêmes titres. Paramita *et al.* (2012) identifient les articles comparables à l'aide des *dumps* [2] et des liens inter-langues sous Wikipedia. Certaines études ne restreignent pas leur collecte à des domaines spécifiques. C'est le cas des travaux de Talvensaari *et al.* (2008) et Leturia *et al.* (2009) qui conçoivent respectivement un crawler thématique et un moteur de recherche pour obtenir des corpus de spécialité de l'Internet.

L'extraction de documents comparables et parallèles des sources exposées ci-dessus requiert une mesure de similarité.

3.2 Mesures de similarité translingue

L'acquisition de documents multilingues est actuellement réalisée au moyen de mesures de similarité. Ces mesures peuvent être de trois sortes : statistiques, sémantiques et éventuellement hybrides.

3.2.1 Mesures statistiques

Des méthodes basées sur le vocabulaire commun et la recherche d'information peuvent être utilisées pour calculer la similarité de textes.

– Selon la méthode basée sur le vocabulaire commun, la similarité de deux textes se mesure à la quantité de mots qu'ils ont en communs. Dans un contexte multilingue, cette mesure repose sur la quantité d'équivalents de traduction que partagent deux textes, obtenue par l'utilisation d'un dictionnaire bilingue ou d'un traducteur automatique. Li et Gaussier (2010) et Su et Babych (2012) développent des mesures de comparabilité en corpus ou textes multilingues à travers l'utilisation de dictionnaires bilingues. Su et Babych (2012) va plus loin dans le prétraitement de données pour contourner les problèmes d'ambiguïté et des différentes formes d'un mot par l'annotation en parties du discours et la stemmatisation.
– Une seconde approche consiste dans la conversion de documents sources en requêtes au moyen de techniques d'extraction de mots clés. Yapomo *et al.* (2012) et Talvensaari *et al.* (2007) utilisent respectivement les mesures TF-IDF [3] et [4] pour obtenir des listes de mots clés représentatives de textes sources. Les documents pertinents sont ceux dont les vecteurs se rapprochent des vecteurs de requêtes. Les requêtes sont traduites dans la langue des documents cibles pour une similarité translingue.

Ces techniques sont certes les moins coûteuses par l'utilisation principale de statistiques de corpus mais l'exclusion d'informations sémantiques soulève plusieurs problèmes. Puisque la similarité est calculée principalement sur la base de mots identiques, synonymie et paraphrase ne sont généralement pas prises en compte. Les limites de cette méthode sont exacerbées dans un contexte multilingue avec la traduction de textes ou de requêtes. Le calcul de la similarité translingue

2. `http://dumps.wikimedia.org/`
3. Term frequency-inverse document frequency (Ramos, 2003)
4. Relative Average Term Frequency (Pirkola *et al.*, 2002)

peut être affecté par le mauvais choix de traductions candidates dans un dictionnaire. De plus, la couverture limitée des dictionnaires/systèmes de traduction en termes de mots nouveaux, de spécialité, ou encore d'unités polylexicales représente aussi un inconvénient considérable. Les mesures sémantiques sont une alternative à cette méthode.

3.2.2 Mesures sémantiques

Plusieurs études exploitent la structure de ressources sémantiques (Leacock et Chodorow, 1998; Jiang et Conrath, 1997) pour calculer la similarité de mots. Corley et Mihalcea (2005) proposent une adaptation de ces méthodes de mesure de la similarité lexicale monolingue à partir de WordNet à la similarité de segments textuels. Les segments textuels sont représentés par leurs mots pleins groupés en catégories grammaticales dont la similarité est mesurée par WordNet. La valeur de similarité entre deux phrases est obtenue par la moyenne des scores de similarité des paires de mots de même catégorie. La spécificité de mots est aussi prise en compte.

A notre connaissance, Il existe peu ou pas d'études utilisant cette approche pour une similarité translingue. Dans cet objectif, une ressource multilingue, à l'exemple de global wordNet [5] ou BabelNet [6] peut être utilisée. L'efficacité de cette méthode est étroitement liée à la qualité de la ressource sémantique utilisée. La structure de WordNet en hiérarchies différentes pour chaque partie du discours limite la portée des similarités qui ne peuvent être qu'intracatégorielles. De plus, la faible composition des hiérarchies autres que celle des noms remet en cause la fiabilité des valeurs de similarité obtenues pour des mots appartenant aux autres catégories grammaticales (verbes, adjectifs et adverbes). Aussi, dans une ressource multilingue, le problème du déséquilibre entre les sous-réseaux sémantiques des différentes langues impliquées se pose. L'absence d'informations sur le contexte pourrait également fausser la liaison des mots de documents aux concepts d'une ressource sémantique : d'où l'introduction de l'approche hybride.

3.2.3 Mesures hybrides

Les recherches s'orientent également vers des mesures de similarité hybrides qui par l'utilisation d'informations en corpus et d'une ressource sémantique, tirent avantage de chacune des approches ci-dessus.

Partant de l'hypothèse selon laquelle le sens d'un mot se détermine en contexte, Mohammad *et al.* (2007) proposent une méthode de calcul de la distance sémantique translingue des mots à travers la comparaison de leurs *profils distributionnels de concepts*. Les profils distributionnels de concepts sont composés des sens de mots non-ambigüs environnants qui permettent d'inférer le sens du mot cible. Pour calculer la similarité de paires de mots allemands, des profils distributionnels de concepts sont construits pour ces paires à partir de leur contextes d'occurrence et d'un thesaurus anglais. Le corpus allemand est mis en correspondance avec le thesaurus anglais à travers un lexique bilingue. Nous pensons que l'adaptation de cette méthode à la similarité de segments plus larges que le mot fournirait de bon résultats. L'utilisation unique d'un thesaurus multilingue sans un recours à des lexiques multilingues est aussi envisageable. Ainsi, Steinberger *et al.* (2002) exploitent les descripteurs du thesaurus multilingue EUROVOC [7]. Ils calculent par ce moyen la

5. http://www.globalwordnet.org
6. http://lcl.uniroma1.it/babelnet/
7. http://eurovoc.europa.eu/

similarité de documents de différentes langues à partir de leurs représentations conceptuelles. Un inconvénient de cette approche est l'effort considérable consacré à l'annotation manuelle d'une collection d'apprentissage en concepts du thesaurus pour l'annotation automatique de nouveaux documents à comparer.

Dans notre but de classifier des documents multilingues, nous nous orientons vers cette approche. Sa nature hybride réside dans le fait qu'elle permette de comparer des documents à l'aide d'une ressource sémantique et d'informations en corpus sur les mots dans les documents à comparer. La mesure de similarité ainsi définie est le principal élément pris en compte par les techniques de construction de corpus multilingues.

3.3 Méthodes de construction de corpus multilingues

Nous abordons dans cette section les principales approches d'acquisition et de structuration de documents multilingues. L'approche Web que nous aborderons est le crawling thématique. Celles pouvant être réalisées indépendamment du Web sont la RII et le clustering.

3.3.1 Le crawling thématique

Une méthode d'extraction de corpus du Web est le crawling qui consiste à utiliser les liens entre les pages pour collecter les documents. Les crawlers thématiques ont été développés pour identifier les sections du Web pertinentes par rapport à un thème donné . Talvensaari *et al.* (2008) utilisent cette méthode pour compiler un corpus comparable de spécialité anglais-espagnol-allemand. Les données de départ sont un ensemble d'URLs reflétant un sujet donné. Les pages correspondantes sont extraites et celles dont les liens figurent dans ces pages initiales sont également visités et prises en compte si le lien thématique avec les pages initiales peut être établi. L'identification du domaine dans les pages candidates se fait sur la base de terminologies collectées séparément pour chaque langue.

Puisque nous n'envisageons pas de développer des techniques d'extraction de documents de l'Internet nous nous intéressons aux approches indépendantes du Web ci-dessous.

3.3.2 Recherche d'information interlingue

Partant de collections de différentes langues, l'approche de RII est aussi utilisée pour collecter des textes comparables. Elle consiste en l'obtention à partir d'une collection source, de mots clés qui sont ensuite traduits et utilisés comme des requêtes exécutées sur la collection cible pour obtenir les documents souhaités. Talvensaari *et al.* (2007) proposent une approche de RII pour la construction d'un corpus comparable suédois-anglais. Les mots clés sont extraits des documents sources par la RATF. Leurs traductions sont exécutées comme requêtes sur la collection cible par le système de recherche d'information Indri qui fait partie du projet Lemur [8]. Les documents obtenus avec cette technique sont appariés ou classés en fonction de leurs scores de similarité avec un document source alors que la méthode de clustering dans la section suivante permet d'obtenir des *clusters* ou groupes de documents.

8. www.lemurproject.org

3.3.3 Clustering de documents

Le clustering de documents se définit comme la répartition d'un ensemble de textes dans des groupes selon leurs traits de similarité sans connaissances a priori. Les documents ayant des caractéristiques communes devraient apparaître dans le même *cluster* (Montalvo *et al.*, 2006). Réciproquement, les documents non- ou peu similaires devraient appartenir à des clusters distincts.

Pour créer des clusters de documents, Li *et al.* (2011) utilisent l'approche agglomérative ascendante. Ils obtiennent des clusters bilingues à partir d'une partie d'un corpus initial. Cette partie regroupe des textes au-dessus d'un seuil minimum de similarité qui serviront à former le corpus comparable. La même procédure est reproduite sur la partie restante du corpus initial par l'intégration de données externes et la création éventuelle de nouveaux clusters. Ertöz *et al.* (2003) utilisent quant à eux l'algorithme de clustering *Shared Nearest Neighbour (SNN)*. Selon cette méthode, deux documents ont plus de chance d'appartenir au même cluster s'ils ont en commun un nombre élevé de voisins. Ils la comparent à la méthode de *K-means* selon laquelle un document appartient à un cluster s'il est proche d'un nombre moyen de documents dans ce cluster (Ertöz *et al.*, 2003).

Nous privilégions le clustering qui va plus loin qu'un simple appariement ou alignement de documents similaires en organisant les documents en groupes.

4 Évaluation

4.1 Évaluation intrinsèque

La valeur d'une méthode de compilation de corpus multilingues peut être estimée à la qualité des données qui en résultent. Il s'agit d'une évaluation intrinsèque. La méthode courante est la comparaison des scores de similarité attribués automatiquement à ceux attribués manuellement. Une échelle de similarité est alors utilisée pour le jugement humain (voir section 2.1.2). Une corrélation faible entre ces deux types de résultats signifie généralement une mauvaise performance du système automatique étant donné que le jugement humain tient lieu de référence.

Pour évaluer leur méthode de calcul de la similarité prenant uniquement en compte les titres des articles, Aker *et al.* (2012) comparent la qualité des alignements obtenus avec celle des alignements produits lorsque le contenu entier de l'article est utilisé. Ils examinent en outre la correspondance entre les résultats de ces méthodes automatiques et la norme que sont les résultats humains. Li et Gaussier (2010) réalisent l'ensemble de l'évaluation automatiquement contournant ainsi l'effort d'annotation manuelle. Le corpus de référence est constitué par des corpus dont la comparabilité est graduellement réduite par l'import de données externes. La corrélation entre les scores de similarité des documents du corpus de référence ainsi construit et ceux obtenus automatiquement est calculée par le coefficent de Pearson. Steinberger *et al.* (2002) utilisent comme critère d'évaluation la capacité de leur méthode à identifier des documents parallèles en leur attribuant les scores de similarité les plus élevés.

4.2 Évaluation extrinsèque

Nous avons vu en section 2.2, les applications des corpus multilingues en TAL. La qualité des données multilingues obtenues est déterminée par leur apport dans ces applications. C'est le cas de l'étude de Talvensaari *et al.* (2007) dans laquelle le corpus comparable obtenu est utilisé comme un thésaurus de similarité accompagné d'un outil de traduction pour améliorer la traduction des requêtes et par ricochet la performance d'un système de RII. Bin *et al.* (2010) entraînent et testent un système de traduction automatique avec un corpus de phrases parallèles obtenu à partir d'un corpus comparable. Li *et al.* (2011) évaluent la qualité du corpus comparable obtenu dans l'application d'extraction de lexiques bilingues. Ils examinent en outre l'apport des lexiques bilingues obtenus en RII.

Des variantes de ces méthodes d'évaluation sont dans un cadre intrinsèque, la comparaison des résultats de plusieurs méthodes automatiques au jugement humain sur les mêmes données (Mihalcea *et al.*, 2006). Pour une évaluation extrinsèque, cela reviendrait à utiliser les données multilingues obtenues d'une même source par différentes techniques de similarité dans une même application et à comparer leurs apports.

Nous avons abordé le sujet de la construction de corpus multilingues en passant en revue quelques principes théoriques et les principales méthodes développées dans ce domaine. Il convient à présent de situer l'apport envisagé dans cette état de l'art.

5 Contributions envisagées

Dans cette partie, nous présentons les limites de la littérature et soulignons quelques perspectives futures notamment celles que nous envisageons de réaliser.

Comme nous l'avons observé en section 2.1.1, la comparabilité se définit dans les limites d'une application. En extraction de lexiques bilingues, elle se mesure généralement à la quantité de vocabulaire que des documents ont en commun (Li *et al.*, 2011). La comparabilité en termes de vocabulaire est-elle suffisante ? Dans un cadre de spécialité où terminologie et vocabulaire n'ont pas la même préséance, l'injection de connaissances du domaine ne serait-elle pas nécessaire à la comparabilité ? Nous prévoyons d'évaluer cette hypothèse de la comparabilité et éventuellement de l'affiner dans notre objectif de concevoir une mesure de similarité permettant d'obtenir des données hautement homogènes. Les mesures de similarité textuelle ont été largement explorées dans un cadre monolingue avec des méthodes basées sur des statistiques de corpus ou sur la structure de ressources ontologiques. Afin d'élaborer une mesure de similarité textuelle translingue et hautement sémantique, nous partirons de corpus multilingues faiblement comparables pour en extraire des sous corpus de meilleure qualité. Notre approche consistera dans la représentation conceptuelle de documents par des descripteurs d'un thesaurus en s'aidant des contextes d'occurrences des mots dans ces documents. Ceci permettra une meilleure attribution des concepts aux documents. Le calcul de la similarité textuelle se fera entre ces représentations. La prise en compte de critères supplémentaires permettra de parfaire la mesure de similarité pour une meilleure classification de documents. Nous adoptons comme approche de construction de corpus multilingues, le clustering qui nous permettra de former des sous corpus ou clusters à partir de notre collection de départ. Les techniques existantes de clustering ne traitent généralement pas de la mise à jour des clusters. Comment permettre l'intégration continuelle de nouveaux

textes à la base ? Le problème de clustering du flux de données restant ouvert, nous pensons que le clustering incrémental (Kurtz, 2012) serait une solution appropriée. Le corpus multilingue obtenu par cette méthode sera évalué en extraction de lexiques, plus précisément de néologismes multilingues. A notre connaissance, aucune étude dans le domaine n'a été entreprise dans cet objectif.

6 Conclusion

Dans cet article, nous avons abordé la construction de corpus multilingues sous plusieurs aspects. Nous avons pu constater que la notion de comparabilité et les applications de corpus multilingues en traitement automatique des langues sont étroitement liées. En effet, la définition de la comparabilité devrait se limiter dans un cadre applicatif donné. Les mesures de similarité textuelles ont généralement suivies les approches statistiques et sémantiques utilisées en majorité dans un contexte monolingue. Les approches hybrides multilingues sont un domaine de la similarité textuelle peu exploré. Pour ce qui est des techniques de clustering de documents existantes, entre autres les méthodes agglomérative hiérarchique, k-means et SNN, elles effectuent une classification ponctuelle et ne résolvent pas le problème de l'ajout permanent de textes à une base. Au vu des limites identifiées, nous prévoyons de fournir une description plus fine de la comparabilité pour l'extraction de lexiques multilingues. Nous envisageons également d'adapter une/plusieurs des méthodes de classification existante(s) au clustering incrémental. La mesure hybride développée déterminera la similarité de textes à travers leurs représentations conceptuelles pour l'application d'extraction de néologismes multilingues.

Références

Aker, A., Kanoulas, E. et Gaizauskas, R. (2012). A Light Way to Collect Comparable Corpora from the Web. *In Proceedings of LREC 2012*, pages 21–27, Istanbul, Turquie.

Baradaran Hashemi, H., Shakery, A. et Faili, H. (2010). Creating a Persian-English Comparable Corpus. *In Proceedings of the 2010 international conference on Multilingual and multimodal information access evaluation*, pages 27–39, Padoue, Italie.

Bekavac, B., Osenova, P., Simov, K. et Tadic, M. (2004). Making Monolingual Corpora Comparable : a Case Study of Bulgarian and Croatian. *In Proceedings of the 4th Language Resources and Evaluation Conference*, pages 1187–1190, Lisbonne, Portugal.

Bin, L., Jiang, T., Chow, K. et Benjamin K., T. (2010). Building a Large English-Chinese Parallel Corpus from Comparable Patents and its Experimental Application to SMT. *In Proceedings of the 3rd Workshop on Building and Using Comparable Corpora*, pages 42–49, La Valette, Malte.

Blancafort, H., Daille, B., Gornostay, T., Heid, U., Sharoff, S. et Méchoulam, C. (2010). TTC : Terminology Extraction, Translation Tools and Comparable Corpora. *In Proceedings of EURALEX 2010*, pages 263–268, Leeuwarden/Ljouwert, Pays-Bas.

Braschler, M. et Schäuble, P. (1998). Multilingual Information Retrieval Based on Document Alignment Techniques. *In Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries*, pages 183–197, Heraklion, Crète, Grèce.

CORLEY, C. et MIHALCEA, R. (2005). Measuring the Semantic Similarity of Texts. *In Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan, USA.

ERTÖZ, L., STEINBACH, M. et KUMAR, V. (2003). Finding Topics in Collections of Documents : A Shared Nearest Neighbor Approach. *Clustering and Information Retrieval*, 11:83–103.

FUNG, P. et YEE, L. Y. (1998). An IR approach for Translating New Words from Nonparallel, Comparable Texts. *In Proceedings of the 17th International Conference on Computational linguistics-Volume 1*, pages 414–420.

GOEURIOT, L., MORIN, E. et DAILLE, B. (2009). Compilation of Specialized Comparable Corpora in French and Japanese. *In Proceedings of the 2nd Workshop on Building and Using Comparable Corpora : from Parallel to Non-parallel Corpora*, pages 55–63, Suntec, Singapore.

ION, R. (2012). PEXACC : A Parallel Sentence Mining Algorithm from Comparable Corpora. *In Proceedings of the Eight International Conference on Language Resources and Evaluation*, pages 2181–2188, Istanbul, Tuquie.

ION, R., TUFIS, D., BOROS, T., CEAUSU, A. et STEFANESCU, D. (2010). On-Line Compilation of Comparable Corpora and their Evaluation. *In FASSBL7*, pages 29–33, Dubrovnik, Crotia.

JIANG, J. J. et CONRATH, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *In Proceedings of International Conference Research on Computational Linguistics (ROCLING X), 1997*, Taipei, Taiwan.

KURTZ, C. (2012). Une distance hiérarchique basée sur la sémantique pour la comparaison d'histogrammes nominaux. *In Actes de Conférence Internationale Francophone sur l'Extraction et la Gestion de Connaissance*, pages 77–88, Bordeaux, France.

LEACOCK, C. et CHODOROW, M. (1998). Combining Local Context and WordNet Similarity for Word Wense Identification. *In WordNet : An electronic lexical database*, page 265–283. Fellbaum, C., Cambridge, MA, MIT Press édition.

LETURIA, I., SAN VICENTE, I. et SARALEGI, X. (2009). Search Engine Based Approaches for Collecting Domain-specific Basque-English Comparable Corpora from the Internet. *In Proceedings of the Fifth Web as Corpus Workshop*, pages 53–61, Donostia-San Sebastian, Basque Country, Spain.

LI, B. et GAUSSIER, E. (2010). Improving Corpus Comparability for Bilingual Lexicon Extraction from Comparable Corpora. *In Proceedings of the 23rd International Conference on Computational Linguistics*, pages 644–652, Beijing, China.

LI, B., GAUSSIER, E., MORIN, E. et HAZEM, A. (2011). Degré de comparabilité, extraction lexicale bilingue et recherche d'information interlingue. *In 18ème conférence sur le Traitement Automatique des Langues Naturelles*, Montpellier, France.

MCENERY, A. M. et XIAO, R. Z. (2007). Parallel and Comparable Corpora : what are they up to ? *In Incorporating Corpora : Translation and the Linguist*. Anderman, G. & Rogers, M., Clevedon, UK, Multilingual Matters édition.

MIHALCEA, R., CORLEY, C. et STRAPPARAVA, C. (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *In Proceedings of the 21st national conference on Artificial intelligence*, volume 1, pages 775–780, Boston, MA, USA.

MOHAMMAD, S., GUREVYCH, I., HIRST, G. et ZESCH, T. (2007). Cross-lingual Distributional Profiles of Concepts for Measuring Semantic Distance. *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 571–580, Prague, République tchèque.

MONTALVO, S., MARTINEZ, R., CASILLAS, A. et FRESNO, V. (2006). Multilingual Document Clustering : an Heuristic Approach Based on Cognate Named Entities. *In Proceedings of the 21st International Conference on Computational Linguistics*, volume 44, pages 1145–1152, Sydney, Australie.

PARAMITA, M., CLOUGH, P., AKER, A. et GAIZAUSKAS, R. (2012). Correlation between Similarity Measures for Inter-Language Linked Wikipedia Articles. *In Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 790–797, Istanbul, Tuquie.

PIRKOLA, A., LEPPÄNEN, E. et JÄRVELIN, K. (2002). The RATF Formula (Kwok's formula) : Exploiting Average Term Frequency in Cross-language Retrieval. *Information Research*, 7(2):7–2.

RAMOS, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. *In Proceedings of the First Instructional Conference on Machine Learning*, Piscataway, NJ USA.

RESNIK, P. (1999). Mining the Web for Bilingual Text. *In Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 527–534, College Park, Maryland, USA.

SKADIŅA, I., AKER, A., GIOULI, V., TUFIŞ, D., GAIZAUSKAS, R., MIERIA, M. et MASTROPAVLOS, N. (2010). A Collection of Comparable Corpora for Under-resourced Languages. *In Proceedings of the Fourth International Conference Baltic HLT*, pages 161–168, Riga, Latvia.

STEINBERGER, R., POULIQUEN, B. et HAGMAN, J. (2002). Cross-lingual Document Similarity Calculation Using the Multilingual Thesaurus Eurovoc. *In Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics*, pages 415–424, Mexico, Mexique.

SU, F. et BABYCH, B. (2012). Measuring Comparability of Documents in Non-parallel Corpora for Efficient Extraction of (Semi-)Parallel Translation Equivalents. *In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 10–19, Avignon, France.

TALVENSAARI, T., LAURIKKALA, J., JÄRVELIN, K., JUHOLA, M. et KESKUSTALO, H. (2007). Creating and Exploiting a Comparable Corpus in Cross-language Information Retrieval. *ACM TOIS*, 25(1):4.

TALVENSAARI, T., PIRKOLA, A., JÄRVELIN, K., JUHOLA, M. et LAURIKKALA, J. (2008). Focused Web Crawling in the Acquisition of Comparable Corpora. *IR*, 11(5):427–445.

UTSURO, T., HORIUCHI, T., CHIBA, Y. et HAMAMOTO, T. (2002). Semi-automatic Compilation of Bilingual Lexicon Entries from Cross-lingually Relevant News Articles onWWW News Sites. *In Proccedings of the 5th Conference of the Association for Machine Translation in the Americas (AMTA) : From Research to Real Users*, pages 165–176, Tiburon, CA, USA.

YAPOMO, M., CORPAS, G. et MITKOV, R. (2012). CLIR- and Ontology-based Approach for Bilingual Extraction of Comparable Documents. *In Proceedings of the 5th Workshop on Building and Using Comparable Corpora*, pages 121–125, Istanbul, Turquie.

Développement de ressources pour l'entraînement et l'utilisation de l'étiqueteur morphosyntaxique TreeTagger sur l'arabe

Dhaou Ghoul[1]
(1) STIH, 1, rue Victor Cousin 75005 Paris
Dhaou.Ghoul@gmail.com

RÉSUMÉ

Dans cet article, nous présentons les étapes du développement de ressources pour l'entraînement et l'utilisation d'un nouvel outil de l'étiquetage morphosyntaxique de la langue arabe. Nous avons mis en œuvre un système basé sur l'étiqueteur stochastique *TreeTagger*, réputé pour son efficacité et la généricité de son architecture. Pour ce faire, nous avons commencé par la constitution de notre corpus de travail. Celui-ci nous a d'abord servi à réaliser l'étape de segmentation lexicale. Dans un second temps, ce corpus a permis d'effectuer l'entrainement de *TreeTagger*, grâce à un premier étiquetage réalisé avec l'étiqueteur ASVM 1.0, suivi d'une phase de correction manuelle. Nous détaillons ainsi les prétraitements requis, et les différentes étapes de la phase d'apprentissage avec cet outil. Nous terminons par une évaluation sommaire des résultats, à la fois qualitative et quantitative. Cette évaluation, bien que réalisée sur un corpus de test de taille modeste, montre que nos premiers résultats sont encourageants.

ABSTRACT

Development of resources for training and the use of the tagger TreeTagger on Arabic

In this paper, we present the steps of the development of resources for training and the use of a new tool for the part-of-speech tagging of Arabic. We implemented a tagging system based on *TreeTagger*, a generic stochastic tagging tool, very popular for its efficiency. First of all, we began by gathering a working corpus, large enough to ensure a general linguistic coverage. This corpus has been used to implement the tokenization process, as well as to train *TreeTagger*. We first present our method of tokenization, then we describe all the steps of the preprocessing and training process, using ASVM 1.0 to yield a raw POS tagging that was subsequently manually corrected. Finally, we implemented a straightforward evaluation of the outputs, both in a quantitative and qualitative way, on a small test corpus. Though restricted, this evaluation showed really encouraging results.

MOTS-CLÉS : TALN, langue arabe, corpus d'apprentissage, étiquetage morphosyntaxique, segmentation de l'arabe, arbre de décision, lexique, jeux d'étiquette, TreeTagger, ASVM 1.0.
KEYWORDS: NLP, Arabic language, training corpus, POS tagging, tokenization, decision tree, lexicon, tagsets, *TreeTagger*, ASVM1.0.

1 Introduction

De nos jours, la langue arabe est de plus en plus utilisée sur le Web. On peut y trouver de nombreux ouvrages que les auteurs ont décidé de rendre publics. Par ailleurs, il existe de nombreux logiciels traitant la langue naturelle qui facilitent la recherche et la consultation des documents électroniques. La réalisation de nouvelles applications en traitement automatique de la langue (TAL) pour l'arabe nécessite en premier lieu de développer un système d'étiquetage performant et robuste.

L'étiquetage morphosyntaxique d'une langue est un processus qui consiste à ajouter aux mots des informations morphologiques concernant leurs catégories morphosyntaxiques ou parties du discours - cette opération étant parfois accompagnée d'une lemmatisation lorsque les formes fléchies sont ramenées à leur forme canonique (lemme). Selon (Laporte , 2000) : « *l'analyse morphosyntaxique est l'ensemble des techniques qui concourent à passer d'un texte brut, exempt d'informations linguistique, à une séquence des mots étiquetés par des informations linguistiques* ». Pour la langue arabe, l'étiquetage reste toujours une étape complexe à aborder à cause des ambiguïtés lexicales des unités. L'étiquetage d'une langue donnée est principalement basé sur deux types d'approche : étiquetage à base de règles et étiquetage statistique basé sur des corpus. L'outil *TreeTagger* que nous avons utilisé dans notre travail concerne cette dernière catégorie de système, et a recours à des modèles probabilistes (modèles de chaine de Markov cachées HMM et arbres de décision).

TreeTagger a déjà été mis en œuvre sur plusieurs langues (anglais, français, allemand, italien, néerlandais, espagnol, bulgare, russe, grec, portugais, chinois, swahili), mais pas sur l'arabe à notre connaissance. L'objectif de ce travail est d'adapter *TreeTagger* à la langue arabe afin de disposer d'un système d'étiquetage morphosyntaxique générique et gratuit.

Cet article est organisé comme suit : la section 2 présente quelques étiqueteurs existants en arabe, la section 3 décrit le principe de segmentation de notre système ainsi que les données utilisées, la section 4 présente le processus de l'étiquetage en se basant sur *TreeTagger*. Les résultats obtenus et les perspectives de ce travail feront l'objet de la section 5.

2 Etat de l'art

Reconnaître la catégorie morphosyntaxique d'un mot dans un contexte est une tâche non triviale du traitement automatique de la langue écrite. En effet rendre une machine capable d'identifier la catégorie d'un mot exige de mettre en œuvre des méthodes sophistiquées, en particulier pour les mots ambigus, c'est-à-dire susceptibles d'appartenir à plusieurs catégories différentes. Les systèmes automatiques dédiés à cette tâche sont appelés des *étiqueteurs* morphosyntaxiques (*part-of-speech tagger*, en anglais). Au contraire de langues comme l'anglais ou le français, l'analyse morphosyntaxique de l'arabe est une étape particulièrement difficile à cause d'importantes ambiguïtés graphiques et de la présence d'agglutinations. De nombreux travaux ont été effectués dans ce domaine en se basant sur des approches différentes. Nous en mentionnons ici quelques uns :

Aramorph est un analyseur distribué par le *Linguistic Data Consortium* (LDC), qui permet de segmenter les mots en trois parties (préfixe racine suffixe). Il utilise un lexique et des règles orthographiques sont encodées directement dans le lexique, spécifiées en termes de règles générales qui interagissent pour réaliser la sortie. Ce système est réalisé à partir d'une base

de données qui contient 598 préfixes, 906 suffixes et 78 839 racines (Buckwalter, 2002). Cette base est complétée par trois tables de comptabilité utilisées pour faire la combinaison entre préfixe et racine (2 435 entrées), suffixe et racine (1 612 entrées) et préfixe et suffixe (1 138 entrées). L'algorithme d'analyse est assez simple puisque toutes les décisions difficiles sont codées directement dans le lexique et les tableaux de compatibilités, c'est-à-dire que le lexique contient toutes les segmentations possibles des mots sous la forme « préfixe racine suffixe » (N.Habash, 2004). Cependant, pour la forme agglutinée d'un mot, les segmentations ne sont valables que si les différents composants existent dans le lexique.

Les points faibles de cet analyseur se résument ainsi (M.Attia, 2006) : tous les lemmes sont listés manuellement et tous les lexèmes des formes fléchies associées sont énumérés, ce qui finit par augmenter le coût de maintenance du lexique ; il y a un problème au niveau du traitement des proclitiques interrogatifs qui se localisent au début des verbes et des noms (exemples : « أأقول », «أمحمد ») ; seulement 22 verbes sur un total de 9 198, soit 0,002 % ont des formes impératives ; seulement 1 404 verbes sur un total de 9 198, soit 15 % sont conjugués à la voix passive au présent et 110 verbes au passé.

(Diab, Hacioglu et Jurafsky, 2004) ont développé un analyseur syntaxique baptisé ASVM 1.0. Ils ont entrainé leurs modèles d'étiquetage sur le corpus arabe annoté TreeBank en se basant sur 24 étiquettes et en utilisant l'outil « Yamcha » qui utilise les machines à vecteurs de support. Le corpus TreeBank utilisé pour évaluer la première version d'ASVM est composé de 4 519 phrases. Le corpus est distribué comme suit : 4 000 phrases pour l'apprentissage, 119 phrases pour le développement et 400 phrases pour le test. Les résultats obtenus sont de 95,49 % de mots correctement étiquetés. Nous avons analysé les résultats de cet étiqueteur (nous les avons utilisés pour notre application), et nous avons remarqué que la majorité des erreurs sont liées à la mauvaise segmentation de l'article « Al » et à la confusion des noms avec les adjectifs et inversement. L'évaluation de la segmentation (tokens bien segmentées) de cet étiqueteur sur notre corpus a donné un taux très faible de segmentation correcte (46 %).

(Diab, 2009) a réalisé une deuxième version améliorée de cet étiqueteur. L'amélioration se résume dans la phase de segmentation concernant les mots composés (par exemple la séparation de l'article « Al (ال)» et la préposition « b (بِ) »). Les résultats obtenus de la nouvelle version sur les mêmes données d'ASVM1.0 sont plus performants au niveau de la segmentation (99,2 %), et avec une précision de plus de 96 % au niveau de l'étiquetage.

(Bahou, Hadrich Belguith, Ben Hamadou, 2005) ont présenté l'analyseur syntaxique SYNTAXE qui modélise des textes arabes non voyellés (non vocalisés) en se basant sur une grammaire HPSG (*Head-driven Phrase Structure Grammar*) et un lexique sous forme XML. Cet analyseur repose sur trois principales étapes. La première étape permet la génération des matrices attribut/valeur HPSG nécessaires pour l'identification des structures syntaxiques des phrases en cours d'analyse. La deuxième et la troisième étape représentent les étapes de l'algorithme « Chart Parsing HPSG » (Popowich, Vogel, 1990). L'évaluation de cet analyseur a été faite sur un corpus de textes tirés d'un manuel de la 8e année de base de l'enseignement tunisien. Ce corpus, saisi au sein du laboratoire LARIS, contient 650 phrases non voyellées (soit 4050 mots). Parmi ces phrases, 96 contiennent des mots non reconnus par l'analyseur et ont été analysé partiellement. Sur un total de 554 phrases, SYNTAXE est parvenu à analyser correctement 448 phrases (2820 mots) soit 81 %.

MorphArab est un analyseur morphosyntaxique de la langue arabe développé par (Abbes,

2004). D'abord, cet analyseur découpe le mot en pré-base, racine et post-base. Ensuite, il utilise le lexique Dinar (Dictionnaire Informatisé de l'Arabe)[1] pour l'attribution de chaque composant du mot et l'extraction des traits morphosyntaxique correspondants. Ce lexique est composé de 19 457 verbes, 70 702 déverbaux (substantif verbaux « مَصدر », participes actifs et passifs « إسم المَفعول والفَاعل », adjectifs et noms de temps et de lieu), 39 099 noms, 445 mots outils et 1 384 noms propres (Anizi, Dichy, 2009). Il identifie en outre les traits morphosyntaxiques des mots. (Abbes, 2004) a trouvé que le moins ambigu des marqueurs est la racine. L'ajout de nouveaux traits augmente la discrimination dans l'analyse et offre plus de solutions.

La société XEROX a également développé un étiqueteur. La phase de segmentation pour cet analyseur est faite par un transducteur à états finis (Farghaly, Dichy, 2003) en découpant la chaîne d'entrée en unités lexicales qui correspondent à une forme fléchie ou une ponctuation, et en donnant à chaque segment des étiquettes qui représentent le comportement morphologique des unités lexicales et leurs catégories. Cet étiqueteur regroupe 4 930 racines et 400 modèles qui permettent de produire 90 000 lexèmes. Il utilise des règles à large couverture, par contre il génère un taux assez élevé d'ambiguïtés lexicales, et ne traite pas bien la phase de désambiguïsation.

TAGGAR est un analyseur morphosyntaxique spécialement développé pour la synthèse vocale arabe des textes voyellés. Il prend en considération l'ordre de traitement des mots pour minimiser les erreurs d'étiquetage. Le traitement se fait dans l'ordre suivant : analyse des mots outils et des mots spécifiques, analyse des formes verbales et enfin, analyse des formes nominales. Cet analyseur utilise 35 étiquettes grammaticales qui se répartissent en trois grandes familles de catégories : 4 étiquettes pour les particules, 16 étiquettes pour les verbes, et 15 étiquettes pour les noms.

L'évaluation a été faite sur un corpus de 5 563 mots ; TAGGAR a obtenu un taux d'erreur de 2 % sur les étiquettes ce qui a entraîné seulement 1 % d'erreurs sur les frontières de groupes syntaxiques. Près de 98 % des pauses insérées automatiquement sont correctement placées (Zemirli, Khabet, 2004).

MORPH2 est un analyseur morphologique basé sur un lexique réduit sous forme XML qui contient 5 754 racines trilittères et quadrilittères (Chaâbaen Kammoun, Hadrich Belguith, Ben Hamadou, 2010) qui correspond à des schémas verbaux et nominaux et un ensemble de règles linguistiques. L'évaluation de cet analyseur a été faite sur un corpus non voyellé qui contient environ 51 404 mots (23 121 différents). Les résultats obtenus en termes de rappel et de précision sont respectivement de 89,77 % et 82,51 %. Cet analyseur prend en entrée un texte en arabe ou une phrase ou un mot pour fournir en sortie toutes les caractéristiques morphosyntaxiques possibles pour chaque mot sans prendre en compte le contexte dans lequel il se présente.

3 Segmentation et données utilisées

Le problème de la segmentation ne se pose pas de la même manière pour toutes les langues. Pour les langues comme l'anglais ou le français, les unités lexicales (tokens) sont dans la plupart des cas reconnaissables par une simple analyse graphique en s'appuyant sur les

[1] http://diinar.univlyon2.fr

caractères séparateurs (espaces, ponctuations, apostrophe, etc.) présents dans les textes.

L'arabe, quant à lui, est en principe monosyllabique, ce qui signifie que chaque syllabe peut être une unité lexicale. Il est possible de former des mots complexes à partir de plusieurs syllabes, ce qui rend difficile le problème de segmentation. La majorité des travaux de segmentation se basent sur des règles qui s'appuient sur des listes de clitiques, préfixes, suffixes et racines (Mars, Zrigui, Belgacem, Zouaghi, Antoniadis, 2008). Ces règles s'appuient sur les principes de constitution d'un mot complexe et son contexte dans la phrase. C'est pourquoi il est difficile d'identifier la racine (unité lexicale) pour les mots qui contiennent des flexions (exemple : les terminaisons des verbes conjugués).

Au cours de notre recherche, nous avons essayé d'élaborer un algorithme de segmentation en nous basant sur des règles qui traitent dans la majorité des cas la forme correcte d'un mot en arabe. Le succès de notre méthode repose essentiellement sur un grand corpus de mots non voyellés segmentés manuellement. Notre algorithme de segmentation est composé de trois modules organisés de manière séquentielle. D'abord, on effectue une segmentation grossière au niveau des espaces et des signes de ponctuations. Ensuite, on examine les tokens[2] ainsi obtenus, et on les compare avec les formes déjà segmentées d'un corpus traité de façon semi-automatique (*cf.* section suivante pour une description du corpus). La segmentation est considérée valide si le token est trouvé dans le corpus. Sinon (*c.-à-d.* si le token est absent du corpus), on recherche, grâce à une expression régulière qui représente la forme complète d'un mot arabe (pré-bases racine post-bases), les éventuelles pré-bases et post-bases attachées à la racine. Cette expression est construite à partir de listes définies à l'avance. Pour chaque pré-base ou post-base identifiée, nous vérifions le statut de la partie restante du mot découpé. Avec cette méthode, nous avons noté qu'il reste des ambiguïtés de découpage pour certains mots qui peuvent se découper de plusieurs façons différentes. Le **Erreur ! Source du renvoi introuvable.**Tableau 1 représente les trois différentes segmentations du mot arabe (**أَلَمهم**) en fonction de son contexte dans la phrase :

Segmentation	Traduction en français
أ + لَم+ هُم	les a-t-il ramassés ?
ألم+ هُم	leurs douleurs
أل+ مُهم	l'important

TABLE 1 – Les différents découpages du mot أَلَمهم

Ce problème reste difficile à résoudre puisque le découpage de ces types de mots dépend obligatoirement du contexte et de sa position dans la phrase. La résolution des cas d'ambiguïté au niveau du découpage reste une tâche non triviale. La qualité de la segmentation dépend de la taille du corpus qui est censé couvrir les mots les plus fréquents en arabe avec leur segmentation correcte.

[2] Ensembles des unités morphosyntaxiques minimales (mot, une partie de mot, clitique...).

4 Adaptation de *TreeTagger* pour l'arabe

4.1 Principe de *TreeTagger*:

TreeTagger est un outil permettant l'étiquetage morphosyntaxique et la lemmatisation. Il a été développé par Helmut Schmid[3] (1994) dans le cadre du projet TC[4]. Il a été utilisé avec succès pour de nombreuses langues (anglais, français, allemand, italien, néerlandais, espagnol, bulgare, russe, grec, portugais, chinois, swahili). Il est adaptable sur toutes les langues en utilisant un lexique et un corpus d'apprentissage manuellement étiquetés. Pour la langue française, (Stein, 2007) a entrainé cet analyseur sur un corpus d'apprentissage contenant 2 685 146 mots et l'a évalué en utilisant un corpus contenant 500 000 mots. Il rapporte un taux de précision de 92,7 % pour l'étiquetage et 97,8 % pour la lemmatisation. *TreeTagger* peut en effet présenter la lemmatisation des mots en plus des étiquettes.

4.1.1 Apprentissage

En général, la phase d'apprentissage des modèles d'étiquetage pour une langue donnée nécessite un corpus d'apprentissage, un lexique de formes fléchies et la liste des étiquettes les plus utilisées pour identifier la catégorie des mots absents du corpus d'apprentissage (classe ouverte). *TreeTagger* utilise des arbres de décision pour l'estimation des paramètres. L'apprentissage effectué par cet outil permet d'évaluer la probabilité d'une transition entre un couple « mot/catégorie» et un autre couple afin de produire un arbre de décision binaire à partir de ces probabilités.

4.1.2 Lexique

En général, l'analyse morphosyntaxique repose sur un lexique contenant les informations sur l'usage grammatical de chaque unité lexicale. Ces informations varient d'un lexique à l'autre. Le lexique joue un rôle important pour l'identification des catégories et du lemme de chaque mot en entrée. Nous avons construit un lexique assez vaste en utilisant la liste de mots proposés par (Buckwalter, 2002) qui était utilisée pour la réalisation de l'étiqueteur *AraMorph*. Nous avons nettoyé cette liste contenant 82 158 racines représentant 38 600 lemmes. Ce nettoyage consiste à éliminer la redondance des mots sur des lignes différentes pour obtenir une forme adaptée à *TreeTagger*. Nous avons ajouté les listes de pré-bases et post-bases à l'entête de notre lexique pour le compléter.

Notons bien que le lexique ne contient pas de chiffres. Par ailleurs, il nous a été difficile de générer automatiquement les lemmes des entrées de notre lexique. Nous avons gardé, pour la plupart des cas, la forme voyellée du mot, et nous avons ignoré les lemmes des pré-bases et post-bases ajoutés manuellement.

4.1.3 Jeux d'étiquettes

Pour gagner de temps, nous avons décidé de prendre le jeu des étiquètes proposés par (Diab, Hacioglu & Jurafsky, 2004) car nous avons utilisé au début ASVM 1.0 pour la préparation de notre corpus de travail, malgré que dont très réduits au niveau de leur nombre en ajoutant une étiquette qui désigne la fin de phrase (étiquette spécifique pour TreeTagger). Ces jeux des étiquettes contiennent 23 étiquettes qui permettent d'identifier les principaux tokens en

[3] http://www.ims.uni-stuttgart.de/~schmid/
[4] http://www.ims.uni-stuttgart.de/projekte/tc/

arabes. Le tableau suivant donne une idée précise sur ces étiquettes :

Tag	Explication	Tag	Explication	Tag	Explication
JJ	Adjective	NNP	Nom propre	PRP$	Pronom possessive
RB	Adverbe	NNPS	Nom propre pluriel	CD	nombre
CC	Coordination	VBP	Verbe à l'imparfait	IN	Subordination
DT	Déterminant	VBN	Verbe passive	UH	Interjection
FW	Mot étranger	VBD	Verbe parfait	PREP	Préposition
NN	Nom singulier	RP	Particule	WP	Pronom relatif
NNS	Nom pluriel	PRP	Pronom personnel	WRB	Wh-adverbe
PUNC	Ponctuation	SENT	Le point de fin de phrase		

TABLE 2 – Listes des étiquettes (Tag).

4.2 Etiquetage

Préalablement le texte à analyser doit être translitteré avec le codage de *Buckwalter* et tokenizé avec notre script de segmentation. TreeTagger a beaucoup de points communs avec les étiqueteurs « n-grammes ». Dans ce type d'approche, on modélise la probabilité d'une séquence de mots en fonction des étiquettes des mots précédents/suivants en se basant sur l'équation suivante :

$$P(w_1 w_2 \dots w_n, t_1 t_2 \dots t_n) = P(t_n | t_{n-2} t_{n-1}) P(w_n | t_n) P(w_1 w_2 \dots w_{n-1}, t_1 t_2 t_{n-1}) \quad (1)$$

Où w_i représente un *mot et* t_i représente une étiquette.

A la différence d'un étiqueteur « n-gramme », *TreeTagger* estime la probabilité de transition à partir d'un arbre de décision binaire généré pendant la phase d'apprentissage. Les nœuds de l'arbre représentent des indices contextuels, et la probabilité d'un trigramme est déterminée par le chemin (de longueur variable) correspondant à travers l'arbre. Par exemple, considérons la séquence trigramme des mots qui ont les étiquettes suivantes « *DT JJ NN* » (avec DT : déterminant, JJ : adjectif et NN : nom). Pour estimer P(NN/DT, JJ), la probabilité d'un nom précédé par un déterminant et un adjectif, on suit le chemin valide en commençant de la racine qui contient l'étiquette *JJ* jusqu'à la feuille qui contient l'étiquette *NN, en passant* par *Tag-2=DT*, et l'on retient la probabilité P=0,8.

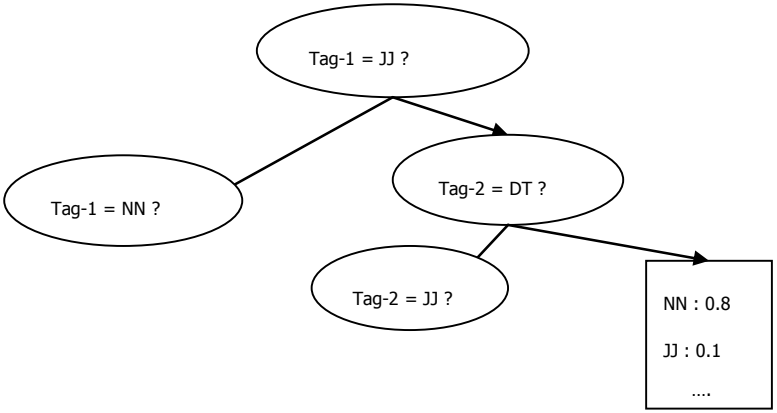


FIGURE 1 – Exemple d'arbre de décision binaire.

Par défaut, pour un texte en arabe segmenté, *TreeTagger* donne une liste de tous les mots avec leurs catégories et leur lemme s'il existe. Un paramétrage de *TreeTagger* permet soit d'attribuer le lemme « unknown » à toutes les formes inconnues, soit de donner la forme elle-même sans lemmatisation. Ici, la phase de segmentation est effectuée par notre propre script et non par les tokenizers génériques de *TreeTagger*. En résumé, ce processus d'étiquetage nécessite un enchaînement de plusieurs phases, comme le montre la figure suivante :

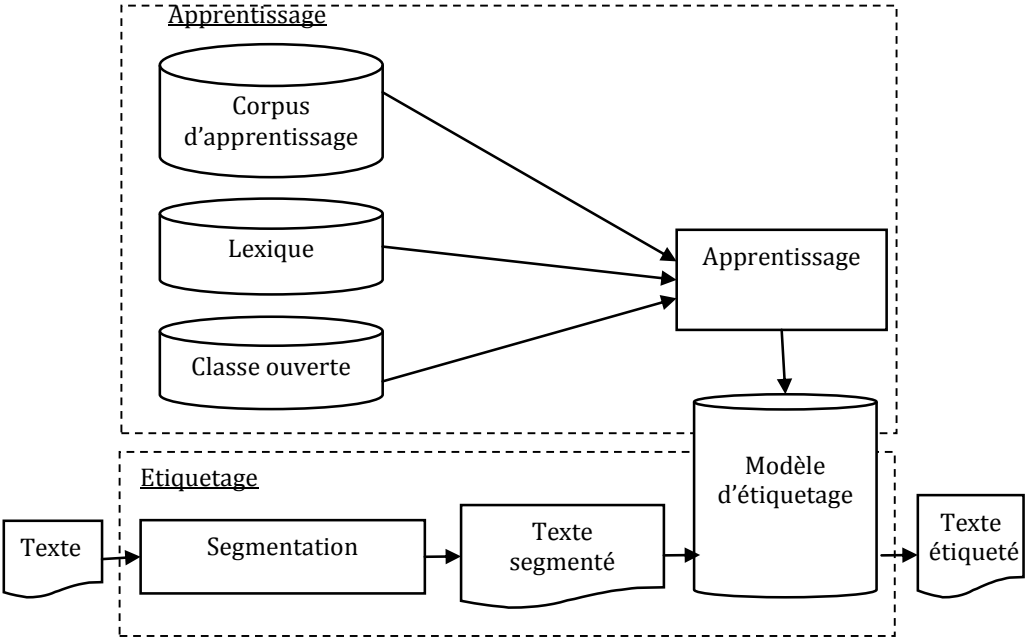


FIGURE 2 – Les différents processus d'étiquetage.

5 Evaluation

5.1 Corpus de travail

Malgré les différentes recherches effectuées sur le traitement automatique de la langue arabe, il nous a été difficile de trouver des ressources toutes faites. C'est pourquoi, nous avons décidé de constituer notre propre corpus de travail. Pour avoir un vocabulaire suffisamment étendu, nous avons utilisé le corpus EASC proposé par (El-Haj , Kruschwitz, Fox., 2010) qui comporte 153 articles répartis sur une dizaine de domaines différents. Notre corpus contient 58 233 mots (21 238 mots différents) repartis sur 2 238 phrases. Comme on l'a dit précédemment, le principe de notre segmentation dépend essentiellement de la comparaison de chaque mot avec les mots pré-segmentés de notre corpus de référence. Pour réaliser cette tâche de pré-segmentation, nous avons suivi les étapes suivantes:

- D'abord, pour éviter tous les problèmes de codage de la langue arabe, et pour faciliter le traitement automatique, nous avons translitteré nos textes selon la table de Buckwalter[5]. Cette translittération, souvent appliquée en TAL, présente l'intérêt de n'utiliser que des caractères ASCII et d'être totalement réversible (c'est-à-dire qu'il est aisé de retrouver le texte original en appliquant la translittération inverse).
- Ensuite, nous avons appliqué l'étiqueteur ASVM 1.0 (étiqueteur gratuit et n'est pas le cas pour ASVM 2.0) réalisé par (Diab, Hacioglu, Jurafsky, 2004) afin d'obtenir une première segmentation selon cet étiqueteur. Comme cet étiqueteur fait beaucoup d'erreurs (segmentation et étiquetage) à ce niveau, il a été nécessaire d'en corriger manuellement les sorties. Nous avons corrigé les 400 premières phrases (soit environ 13 000 mots), en définissant, pour chaque correction, des transformations sous forme d'expressions régulières, que nous avons appliquées à l'ensemble du corpus. A l'issue de ces corrections, la présence de mots mal segmentés ou mal étiquetés devenait de plus en plus rare (pour donner une estimation; après la correction de 400 phrases, une erreur apparaît tous les 200 à 300 mots). Notons bien que, avant la phase de correction et selon la sortie qu'on a obtenue, le taux d'erreur est assez élevé (une erreur apparaît à-peu-près tous les 50 mots).
- Enfin pour compléter notre corpus nous avons ajouté la liste de pré-bases et post-bases les plus utilisées en arabe, donnée par (Abbes, 2004).

5.2 Corpus d'apprentissage et d'évaluation

Pour créer les corpus d'apprentissage et d'évaluation, nous avons pris le corpus de travail déjà utilisé pour la segmentation et nous en avons extrait 234 phrases pour constituer le corpus de test, le reste étant réservé au corpus d'apprentissage. Le tableau 3 illustre la taille des corpus utilisés. Le corpus d'apprentissage contient 52 171 mots non segmentés dont 19 086 mots différents regroupés dans 2 096 phrases. Notons que l'ensemble des mots du corpus est non voyellé. La distribution des 23 étiquettes en question dans le corpus d'apprentissage donne une valeur minimale pour l'étiquette « FW (mot étranger)» (192 fois) et une valeur maximale pour l'étiquette « DT (déterminant) » (11 264 fois) sur une totalité de 78 650 mots étiquetés.

[5] http://www.qamus.org/transliteration.htm

Corpus	Nb. phrases	Nb. mots différents	Nb. mots non-segmentés	Nb. mots segmentés
Apprentissage	2 096	19 086	52 171	78 650
Test	234	3 407	6 029	9 560

TABLE 3 – Statistiques de distribution du corpus pour l'apprentissage et le test.

5.3 Evaluation quantitative

Le corpus de test est constitué d'articles concernant la thématique de l'art. Pour mettre en œuvre l'évaluation, nous avons besoin de réaliser un étiquetage de référence qui contient les phrases de test bien segmentées et avec des étiquettes vérifiées manuellement.

Nous nous sommes limités au problème de l'évaluation de la précision de l'étiquetage réalisé par *TreeTagger*, c'est-à-dire le taux d'étiquetage correct. Cependant, il faut être conscient que ce seul taux ne signifie que peu de chose dans la comparaison entre les systèmes, car la précision de chaque système dépend du mode de segmentation et du jeu d'étiquettes utilisés, ainsi que des données de test utilisées.

Une démarche d'évaluation simple consiste à comparer le résultat de notre étiqueteur avec le corpus de référence. Pour faire cette comparaison, nous avons décidé d'utiliser *Sclite*[6], un outil générique pour l'évaluation des étiqueteurs morphosyntaxiques. Ce logiciel nous a permis de faire l'alignement entre le texte étiqueté et la référence, les segmentations pouvant être différentes.

Pour l'évaluation d'un système d'étiquetage, on peut considérer les éléments suivants :

- Une évaluation des types d'ambiguïté pour apprécier la difficulté de l'étiquetage : le nombre moyen d'étiquettes possibles à assigner à chaque mot, et les types (ou classes) d'ambiguïté, en précisant notamment la fréquence relative dans le corpus test de chacune de ces classes.
- Une évaluation des types d'erreurs : les types d'ambiguïté conduisant le plus fréquemment à des erreurs d'étiquetage, ainsi qu'au mauvais découpage des mots.
- Une évaluation quantitative de la précision de l'étiquetage.

Par ailleurs, nous avons comparé les résultats obtenus avec ceux d'ASVM1.0 (également appelé AMIRA1.0), pris comme baseline. Les deux mesures communément utilisées pour évaluer un système d'étiquetage sont le taux de précision P et celui du rappel R. Tous les mots étant étiquetés, nous avons limité l'évaluation au calcul de la précision en comparant les couples mot/catégorie des phrases étiquetées à ceux des phrases de référence. Nous avons d'abord calculé la précision au niveau de chaque phrase. Pour calculer la précision globale, on calcule ensuite la précision moyenne sur l'ensemble des phrases de test :

$$P_i = \frac{nombre\ des\ couples\ corrects\ obtenus}{nombre\ total\ des\ mots\ bien\ segmentés} \times 100 \qquad (2) \qquad P_{moy} = \frac{\sum_{i=1}^{n} P_i}{n} \qquad (3)$$

[6] http://www.itl.nist.gov/iad/mig/tools/

Où n est le nombre de phrases (n=234), et P_i : la précision de la phrase i.

Ce mode de calcul a pour effet de minimiser l'impact des erreurs qui apparaissent dans les phrases très longues, car il donne une pondération plus importante aux étiquettes des phrases courtes.

Pour obtenir une évaluation comparative précise, nous avons évalué les deux systèmes sur le même corpus de test. Nous avons obtenu un taux de précision moyen égal à 86.5 % contre 60 % pour ASVM1.0. Nous avons également évalué la tâche de segmentation sur le même corpus de test, et obtenu un taux de précision de 93 %. Le tableau ci-dessous résume bien cette évaluation.

	Notre système	ASVM1.0
Étiquetage	86,5 %	60 %
Segmentation	93 %	85 %

TABLE 4 – Evaluation quantitative de notre système et ASVM 1.0.

5.4 Evaluation qualitative

Afin d'avoir une idée plus précise des types d'erreur rencontrés, nous avons procédé à une évaluation qualitative. Nous avons examiné seulement les 50 premières phrases. Ces phrases contiennent 869 mots étiquetés parmi lesquels nous avons trouvé 24 étiquettes fausses c'est-à-dire 2,75 % d'étiquettes erronées.

Nous avons constaté que les erreurs sont dues, en général, à la mauvaise segmentation des mots, à l'absence des mots dans le lexique, ainsi qu'aux cas d'ambiguïté. Le tableau suivant illustre les différents cas de figure pour ces 24 étiquettes erronées :

Phrases	Mots	Etiquettes erronés	Mots mal segmentés	Mots absent du lexique	Mots ambigus aux niveaux lexical et grammatical
50	869	24	11	9	4
pourcentage		2,75 %	1,26 %	1,03 %	0,46 %

TABLE 5 – Les différents cas d'étiquettes erronées sur 50 phrases examinées manuellement.

Pour comparer les résultats de notre étiqueteur et ceux d'ASVM1.0, nous avons choisi, à titre d'illustration, quelques phrases étiquetées par les deux systèmes :

Phrases	Notre étiqueteur	ASVM1.0
لوديفج فان بيتهوفن مؤلف موسيقي ألماني ولد عام	lwdfYj / NNP fAn / NNP bYthwfn / NNP m&lf / NN mwsYqY / JJ >lmAnY / JJ wld / VBN EAm / NN 1770 / CD m / NN fY / IN mdYnp / NN bwn /	***lwdfYj / NN*** fAn / NNP bYthwfn / NNP m&lf / NN ***mwsYqY / NN >lmAnY / NNP*** wld / VBN EAm /

1770 م في مدينة بون.	NNP . / SENT	NN 1770 / CD m / NN fY / IN mdYnp / NN bwn / NNP . / PUNC
يعتبر من أبرز عباقرة الموسيقى في جميع العصور، و أبدع أعمال موسيقية خالدة.	YEtbr / VBN mn / IN >brz / JJ EbAqrp / NN Al / DT mwsYqy / NN fY / IN jmYE / JJ Al / DT Eswr / NNS , / PUNC w / CC >bdE / VBN >EmAlAF / NN mwsYqYp / JJ xAldp /JJ . / SENT	YEtbr / VBN mn / IN >brz / JJ EbAqrp / NN ***AlmwsYqy / NN*** fY / IN jmYE / JJ ***AlEswr; / NN*** w / CC >bdE / VBN >EmAlAF / NN mwsYqYp / JJ xAldp /JJ . / PUNC
لذلك ينصح عادة بأن يتمرن ممن يريد التعلم بالتدريب على إخراج الصوت أولا ومن ثم عندما يستطيع ذلك يبدأ بالتعلم على إخراج الدرجات الصوتية (تمرين الأصابع).	l/IN *lk/WP YnSH/VBD EAdp/NN b/PREP >n/RP Ytmrn/VBN mn/IN YrYd /VBN Al/DT ***tElm/VBN*** b/PREP Al/DT tdrYb/NN Ely/RP <xrAj /NN Al/DT Swt/NN >wlA /JJ w/CC mn/IN vm/RB EndmA/IN YstTYE/VBD *lk/WP Ybd>/VBP b/PREP Al/DT ***tElm/VBN*** Ely/RP <xrAj /NN Al/DT drjAt/NNS Al/DT SwtYp/NN (/PUNC tmrYn /NN Al/DT >SAbE /NNS)/PUNC ./SENT	***l*lk/IN*** YnSH/VBD EAdp/NN b/IN ***>n/IN*** Ytmrn/VBN mn/IN ***YrYd /NN AltElm/NN*** b/IN ***AltdrYb/NN*** Ely/IN <xrAj /NN ***AlSwt/NN*** >wlA /JJ w/CC mn/IN vm/RB EndmA/IN YstTYE/VBD ****lk/DT Ybd>/NN*** b/IN ***AltElm/NN*** Ely/IN <xrAj /NN ***AldrjAt/NNS AlSwtYp/JJ*** (/PUNC tmrYn /NN ***Al>SAbE /NNS***)/PUNC ./PUNC
قدم أول عمل موسيقي و عمره 8 سنوات.	***qdm / NN*** >wl / JJ Eml / NN mwsYqY / JJ w / CC Emr / NN h / PRP 8 / CD snwAt / NNS . / SENT	qdm / VBD >wl / JJ Eml / NN mwsYqY / JJ w / CC Emr / NN h / PRP$ 8 / CD snwAt / NNS . / PUNC

TABLE 6 – Exemples de phrases étiquetées par notre étiqueteur et ASVM1.0.

Si on observe les résultats obtenus par notre étiqueteur sur ces 4 phrases, on remarque bien qu'il analyse les deux premières phrases correctement, par contre, il génère des erreurs au niveau de la troisième et de la quatrième phrase. En général, les erreurs d'étiquetage de notre système viennent soit de la mauvaise segmentation du mot ou de l'absence du mot dans le corpus d'apprentissage, soit de l'ambiguïté graphique du mot liée à l'absence des voyelles (p.ex. qdm ***(قدم)/NN***).

6 Conclusion et perspectives

Dans ce travail, nous avons essayé d'adapter l'outil *TreeTagger* sur la langue arabe. Pour ce faire, nous avons organisé notre travail en trois étapes principales. D'abord, nous avons récolté et préparé toutes les données nécessaires : lexique, jeux d'étiquettes et corpus d'apprentissage. Ensuite, nous avons développé une méthode de segmentation des textes

arabes basée sur corpus pré-segmenté manuellement. Enfin nous avons terminé par une évaluation qualitative et quantitative de notre système. L'évaluation sommaire que nous avons menée indique que notre étiqueteur arabe donne 86 % de précision. Les résultats de l'évaluation quantitative montrent un gain de notre méthode par rapport à l'étiqueteur ASVM1.0. Malgré un corpus d'apprentissage très restreint, ces résultats sont donc encourageants.

Pour le moment notre système ne permet pas de lemmatiser, car notre lexique, encore incomplet, ne contient pas de lemmes : dans nos prochains travaux, nous comptons remédier à cette lacune en ajoutant cette information. De plus, pour obtenir un étiqueteur générique à large couverture, nous envisageons d'augmenter notre corpus d'apprentissage, sur le plan quantitatif, mais aussi de l'enrichir en termes de variété typologique (littéraire, journalistique, scientifique, etc.).

7 Références

ABBES, R. (2004). La conception et la réalistion d'un concordancier. Lyon,ENSSIB/INSA: Thèse de doctorat en sciences de l'information.

ANIZI, M. et DICHY, J. (2009). Assessing Word-form based Search for Information in Arabic: Towards a New. *MEDAR 2and International conference on Arabic Language Ressources & Tools* , Cairo Egypt, pages 12-19 .

ATTIA, M. (2006). An Ambihuity controlled Morphological Analyser for Modern Standard Arabic Modelling Finite State networks. *: Acte de la conférence internationale 'the challenge of arabic for NLP/MT, the Britsh computer society.* London.

BUCKWALTER, T. (2002). Arabic Morphological Analyser version 1.0. Linguistic Data Consortium Catalogue numéro LDC L 49.

BAHOU, Y., HADRICH BELGUITH, L., et BEN HAMADOU, A. (2005). SYNTAXE: Analyseur syntaxique de l'arabe utilisant XML comme outil de stokage . Souuse,Tunisie: *Cinquièmes journés scientifique des jeunes chercheurs en génie électrique et informatique* .

CHAÂBAEN KAMMOUN, N., HADRICH BELGUITH, et BEN HAMADOU, A. (2010). The MORPH2 new version: A Robust morphological analyser for arabic text. MIRACL Laboratory Tunisia .

DIAB, M. (2009). Second Generation AMIRA for Arabic Processing: Fast and Robust Tokinisation, Pos Tagging and Base Phrase Chunking. MEDAR 2nd International conference on Arabic Language Processing & Tools, Cairo Egypt, pages 285-288.

DIAB, M., HACIOGLU, K.,et JURAFSKY, D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. . *HLT-NAACL* , pages 149-152.

EL-HAJ, M ,KRUSCHWITZ, U. et FOX, C. (2010). Using Mechanical Turk to Create a Corpus of Arabic Summaries in the Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages. *workshop held in conjunction with the 7th International Language Resources and Evaluation Conference (LREC 2010)*, Valletta, Malta, pages 36-39.

FARGHALY, A.et DICHY, J. (2003). Roots & Patterns VS Stems plus Grammair-Lexis specification :On what basis should a multilingual lexical database centred on arabic be built? *Acte de la 9ème MT conference, Workshop on Machine translation for semitic language :issues and approaches.* New Orleans, Louisiana, USA.

LAPORTE, E . (2000). Mot et niveau lexical . *jean-marie pierre: Ingenierie de langues* , pages 25-46.

MARS, M., ZRIGUI, M., BELGACEM, M., ZOUAGHI, A. et ANTONIADIS, G. (2008). A Semantic Analyzer for the Comprehension of the Spontaneous Arabic Speech. *International Conference on Computing CORE08, Journal Research in Computing Science (Journal RCS)* , ISSN: 1870-4069, Vol 34, pages 129-140.

POPOWICH, F. et VOGEL, C. (1990). Chart parsing head-driver phrase structure grammar. *Technical Report CSS-IS TR 90-01, CMPT TR 90-01, Simon Fraser University, Burnaby, BC.*

SCHMID, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK, pages 88-96.

STEIN, A. (2007). Part of speech tagging and lemmatisation of Old French. `http://www.uni-stuttgart.de/lingrom/stein/forschung/resource.html`. [consulté le 10/02/2011].

ZEMIRLI, Z. et KHABET, S. (2004). Un analyseur morphosyntaxique destiné à la synthèse vocale de textes arabes voyellés. *JEP-TALN,* Fès.

Détection de polarité d'opinion dans les forums en langues arabe par fusion de plusieurs SVMs

Ziani Amel[1], Tlili Guiassa Yamina[2], Azizi Nabiha[3]
[1] Département d'informatique, Université Badji Mokhtar Annaba (Algérie)
[2]Lri laboratory : laboratoire de recherche en informatique
[3]Labged laboratory : Laboratoire de gestion électronique des documents
Z_amel1911@live.fr, yamina.tlili@univ-annaba.org,nabiha.azizi@univ-annaba.org

RÉSUMÉ

Cet article décrit notre contribution sur la détection de polarité d'opinions en langue arabe par apprentissage supervisé. En effet le système proposé comprend trois phases: le prétraitement du corpus, l'extraction des caractéristiques et la classification. Pour la deuxième phase, nous utilisons vingt caractéristiques dont les principales sont l'émotivité, la réflexivité, l'adressage et la polarité. La phase de classification représente dans notre travail la combinaison des plusieurs classifieurs SVMs (Machine à Vecteur de Support) pour résoudre le problème multi classes. Nous avons donc analysés les deux stratégies de SVM multi classes qui sont : « un contre tous » et « un contre un » afin de comparer les résultats et améliorer la performance du système global.

ABSTRACT

Polarity Opinion Detection in Arabic Forums by Fusing Multiple SVMs

This article describes our contribution on the polarity's detection of opinions in Arabian language by supervised training. Indeed the proposed system consists of three phases: the pretreatment of the corpus, the extraction of the features and the classification. For the second phase, we use twenty features of which the main are emotionalism, the reflexivity, the adressage and the polarity. The phase of classification represents in our work the combination of the several SVMs (Support Vector Machine),to solve the multi class problem. We analyzed the two strategies of the SVMs multi class that are: "one against all" and "one against one" in order to compare the results and to improve the performance of the global system.

MOTS-CLÉS : Fouille d'opinions, apprentissage supervisé, Machine à Vecteur de Support (SVM), combinaison des classifieurs.
KEYWORDS : Opinion Mining, supervised training, Support Vector Machine (SVM), classifiers combination.

1 Introduction

De nos jours plusieurs applications et plateformes sur le web nous permettent de déposer des avis, de partager des sentiments et des opinions sur une variété de sujets. Vue l'importance de ces informations dans plusieurs domaines (politique, commercial ou individuel), il serait important de déterminer l'information subjective contenue dans les

textes.

Mais la détection automatique d'opinions et l'analyse des sentiments sont confrontées à des problèmes qui la distinguent de la recherche thématique traditionnelle, car le sentiment est exprimé de manière très variée et très subtile. La nécessité de traiter automatiquement les opinions se fait donc fortement ressentir.

La détection d'opinions est une tâche qui permet d'extraire les opinions d'un ensemble de documents pertinents pour un sujet donné. Elle est confrontée à des problèmes qui la distinguent de la recherche traditionnelle thématique dont les sujets sont souvent identifiés par des mots clés seulement. La classification du sentiment (polarité) est une sous-tâche de la détection d'opinions. Elle consiste de façon générale à déterminer si l'opinion du document sur le sujet est positive ou négative. Puisque nous avons besoin d'associer des notes à des textes, nous nous intéressons ici uniquement à la classification d'opinions. Deux grands types de méthodes sont utilisés pour cette tâche. Il y a tout d'abord les approches plutôt linguistiques qui consistent à répertorier le vocabulaire porteur d'opinions, puis à établir des règles de classification selon la présence ou l'absence des mots appartenant à ce vocabulaire. Il existe également les approches mettant en œuvre des outils issus du domaine de l'apprentissage automatique. Nous nous intéressons ici uniquement à celles de la deuxième famille, qui sur nos données se sont avérées nettement plus efficaces. Les méthodes utilisées dans ce cadre sont issues de la classification dite supervisée, où un classifieur est appris à l'aide d'exemples de données dont on connaît déjà la classe.

Ainsi, si aucune méthode de classification ne peut satisfaire entièrement aux exigences d'une application envisagée, l'utilisation simultanée de plusieurs méthodes en même temps peut éventuellement permettre d'en cumuler les avantages sans en cumuler les inconvénients. En effet, le comportement de chaque classifieur vis-à-vis de commentaire à classifier, est déterminé à partir des informations différentes représentants les caractéristiques extraites. L'exploitation des différents résultats générer par les classifieurs utilisant une des méthodes de combinaison de classifieurs, aboutir généralement à un commentaire bien classifié. Même si le classifieur est moins performant, la connaissance de son comportement apporte une certaine information utilisable à propos de la vraie classe pendant la combinaison. Donc, le but de la combinaison de classifieurs vise à réduire l'erreur et augmenter la fiabilité de la classification.

En effet, nous avons conçu et implémenté toutes les phases du processus de classification allant de la construction de la base de corpus jusqu'à la classification par combinaison d'expert, tout en passant par la phase d'extraction des caractéristiques. Concernant ces derniers, les caractéristiques que nous avons jugées importantes sont l'émotivité, la subjectivité, la réflexivité et l'adressage. Nous nous intéressons ensuite à la détection de polarité en déterminant si l'opinion est fortement positive, positive, fortement négative, négative ou neutre utilisant une des méthodes de classification. Toutefois, les méthodes les plus présentes dans la littérature, et qui semblent également être les plus performantes dans le domaine de fouille d'opinion sont les machines à vecteurs de supports. Comme notre système doit générer 5 classes, il est considéré comme un problème de classification multi classe, pour cela nous avons adopté les deux stratégies « un-contre-tous » et « un contre un » pour décider laquelle entre ces deux est meilleure dans le domaine de fouille d'opinions. Les SVMs sont jugés être très sensibles au paramètres internes tels que la fonction noyau ; nous avons donc décidé d'analyser le changement de la fonction noyau en générant plusieurs classifieurs SVMs multi-classes associé chacun avec une fonction noyau différente.

Le résultat final sera la combinaison de ces classifieurs afin d'assurer la complémentarité existante entre les différentes fonctions. Pour cela on a utilisé les deux méthodes de fusion : Vote Majoritaire et Vote Pondéré pour générer les résultats finals.

Nous proposons dans cet article une approche de classification d'opinions dans les journaux arabes fondée sur la combinaison parallèle des classifieurs SVM (Support à Vecteur Machine) en utilisant les quatre fonctions (linéaire, polynomiale, gaussienne et tangente) afin de comparer les résultats. Dans un premier temps, nous introduisons le domaine de la fouille d'opinions avec un état de l'art relatif à notre problème. Ensuite, nous présentons les difficultés de la langue arabe et le processus général de la classification d'opinions. Puis, nous décrivons les différentes phases de l'approche proposée, incluant l'ensemble des caractéristiques adoptées. Après nous présentons et discutons les résultats obtenus à travers les expérimentations que nous avons menés. Nous achevons cet article par une conclusion et un ensemble de perspectives.

2 Etat de l'art

Le terme « fouille d'opinions » est utilisé pour évoquer le traitement automatique des opinions, des sentiments et de la subjectivité dans les textes. Ce domaine est connu sous les noms de : opinion mining (Pang et Lee, 2008), sentiment analysis (Liu, 2010), ou encore subjectivity analysis et est souvent associé à un problème de classification sur des textes évaluatifs comme ceux disponibles sur Amazon ou Epinions. Afin de décider de l'orientation d'un document (Turney, 2002), (Wilson et al., 2004) ou de la valeur positive/négative/neutre d'une opinion dans un document (Hatzivassiloglou et McKeown, 1997), (Yu et Hatzivassiloglou, 2003), (Kim et Hovy, 2004).

Le travail de Maurel et Dini en 2009 été caractérisé par une utilisation mixte d'une technologie symbolique fondée sur des règles et d'une technologie statistique reposant sur l'extraction d'une ontologie du domaine, approche dans laquelle la méthode symbolique a un poids plus important (Dini, 2002), (Dini et Mazzini, 2002), (Maurel et al., 2007), (Maurel et al., 2008), (Bosca et Dini, 2009).

Des travaux allant au-delà ont mis l'accent sur la force d'une opinion exprimée où chaque proposition dans une phrase peut avoir un fond neutre, faible, moyen ou élevé (Wilson et al., 2004). Des catégories grammaticales ont été utilisées pour l'analyse de sentiments dans (Bethard et al., 2004) où des syntagmes adjectivaux comme trop riche ont été utilisés afin d'extraire des opinions véhiculant des sentiments. (Bethard et al., 2004) utilisent une évaluation basée sur la somme des scores des adjectifs et des adverbes classés manuellement, tandis que (Chklovski, 2006) utilise des méthodes fondées sur un modèle pour représenter des expressions adverbiales de degré telles que parfois, beaucoup, assez ou très fort.

En 2002, Turney, Pang et coll. encouragent la recherche dans le domaine de sentiment analysis en classant des critiques de cinéma. (Ding et al., 2007) analysent les cooccurrences de mots à l'intérieur d'une phrase puis les cooccurrences entre les phrases. Ils combinent des règles issues des deux échelles pour obtenir un meilleur taux de bonne classification en classification de sentiments. Dans le même genre. (Wilson et al., 2004) ajoutent à la classification selon la polarité, la force de l'opinion exprimée.

En ce qui concerne les études sur la fouille d'opinions en langue arabe, elles n'en sont qu'à leur début, présentons ici quelques travaux : (Almas et Ahmed, 2007), (Abbasi et al., 2008), (Elhawary et Elfeky, 2010), (El halees, 2010), (Rushdi et al., 2011) et(Montassir et al., 2012).

3 La langue arabe

Par ses propriétés morphologiques et syntaxiques la langue arabe est considérée comme une langue difficile à maîtriser dans le domaine du traitement automatique de la langue. L'arabe doit sa formidable expansion à partir du 7eme siècle grâce à la propagation de l'islam et la diffusion du Coran. Les recherches pour le traitement automatique de l'arabe ont débuté vers les années 1970. Les premiers travaux concernaient notamment les lexiques et la morphologie.
Avec la diffusion de la langue arabe sur le Web et la disponibilité des moyens de manipulation de textes arabes, les travaux de recherche ont abordé des problématiques plus variées comme la syntaxe, la traduction automatique, l'indexation automatique des documents, la recherche d'information, le résumé automatique, la fouille d'opinions etc.
Un des aspects complexes de la langue arabe est l'absence des voyelles dans le texte, qui risque de générer une certaine ambiguïté à deux niveaux :

- Sens du mot
- Difficulté à identifier sa fonction dans la phrase, (différencier entre le sujet et le complément).

Ceci peut influencer les fréquences des mots étant donné qu'elles sont calculées après la détection de la racine ou la lemmatisation des mots qui est basée sur la suppression de préfixes et suffixes.
Bien que la lemmatisation soit difficile pour les langues avec des morphologies complexes comme l'arabe, elle est particulièrement importante et utile en particulier dans les systèmes de recherche d'information. Il est suffisant de regrouper les mots qui se ressemblent le plus sans pour autant connaître la racine exacte.
Contrairement à l'anglais, la langue arabe possède un système dérivationnel très riche, et c'est dans cette caractéristique que réside la difficulté de traiter cette dernière.

4 Processus général de la classification d'opinions

Puisque nous avons besoin d'associer des notes à des textes, nous nous intéressons ici uniquement à la classification d'opinions. Deux grands types de méthodes sont utilisés pour cette tâche. Il y a tout d'abord les approches plutôt linguistiques qui consistent à répertorier le vocabulaire porteur d'opinions, puis à établir des règles de classification selon la présence ou l'absence des mots appartenant à ce vocabulaire.

Il existe également les approches mettant en œuvre des outils issus du domaine de l'apprentissage automatique. Dans notre cas, un ou plusieurs classifieurs sont utilisés pour apprendre la polarité des opinions traités, puis ils seront responsables de classifier les opinions inconnues.

Nous nous intéressons dans ce travail à l'application de la deuxième approche avec un module d'extraction des caractéristiques pertinentes. Les méthodes utilisées dans ce cadre sont issues de la classification dite supervisée (ou apprentissage supervisé), où un classifieur

est appris à l'aide d'exemples de données (ici de textes de commentaires) dont on connaît déjà la classe (ici la polarité). Les mots des textes sont alors généralement considérés comme des données indépendantes et équivalentes les unes aux autres, leur sémantique n'étant pas explicitement prise en compte. On peut donner une définition un peu plus formelle du problème de la classification supervisée comme suit :

Définition : soit X un ensemble de données, Y un ensemble d'étiquettes (ou classes) et D un ensemble des représentations des données. Soit $d : X \rightarrow D$, une fonction connue qui associe à chaque donnée $x \in X$ une représentation $d(x) \in D$ et $S \subset D \times Y$ un ensemble de données étiquetées $(d(x), y)$. La classification supervisée consiste à construire en s'appuyant sur l'ensemble S un classifieur, c'est-à-dire une fonction de $D \rightarrow Y$ qui permette de prédire la classe de toute nouvelle donnée $x \in X$, représentée par $d(x) \in D$.

D'après cette définition, trois éléments distincts entrent en jeu dans la classification supervisée :

- Les étiquettes ou classes de prédiction (l'ensemble Y) ;
- Les exemples de données étiquetées, qui constituent le corpus d'apprentissage (l'ensemble S) ;
- Le classifieur ou prédicteur.

De plus, des prétraitements peuvent être appliqués sur les données avant la tâche de classification dans le but d'améliorer ses performances, que ce soit en termes de résultats ou de temps de calcul. (Poirier et al, 2011)

5 Structure générale du système AROPOL (ARabic Opinions POLarity) basée sur la classification supervisée

Nous pouvons résumer notre approche de classification d'opinions par combinaison des SVM dans les journaux en langue arabe par le schéma suivant :

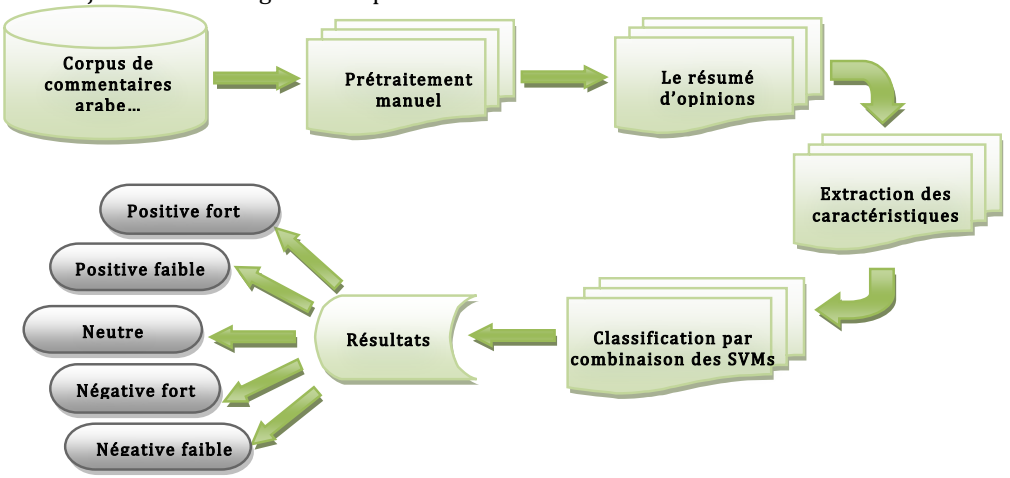


FIGURE 1 – Le Processus de classification d'opinions par l'approche AROPOL

5.1 Le corpus d'apprentissage

La classification supervisée nécessite des exemples (données étiquetées) afin de construire le «corpus d'apprentissage». Ce corpus ayant un impact direct sur l'apprentissage des règles, et par conséquent sur la classification, il est nécessaire que les exemples soient représentatifs de l'ensemble des données.

Cette hypothèse est généralement difficile à vérifier. En classification d'opinions, ou plus généralement en classification de textes, les corpus étudiés sont assez restreints car l'étiquetage des exemples est souvent effectué à la main. Ceci entraîne un coût élevé et ne permet donc pas l'obtention d'un gros corpus d'apprentissage. (Poirier et al, 2011)

Pendant notre travail, nous avons utilisés un corpus de cent cinquante commentaires sur des articles, recueillis à partir des journaux arabes algériens disponibles sur le net (Eshorouk الشروق, Akher saa آخرساعة etc.). L'ensemble touchait plusieurs thèmes différents (économiques, politiques, etc.). Notre but était d'effectuer une classification supervisée pour pouvoir déterminer la polarité des commentaires: positive fort, positive faible, neutre, négative fort ou négative faible. C'est pour cela on a appliqué un ensemble des prétraitements manuels sur ce corpus.

Exemple de commentaire :

Nous présentons un exemple de chaque type d'opinion que nous devons classifier.

	Polarité	Les commentaires en français	Les commentaires en arabe
1	Positif faible	Merci pour ce que tu as écris Mr amine zaoui celui qui dirige ministère de la culture en Algérie est celui que le dépense sans dignité.	شكراعلماكتبت سيد الأمين الزاوي إن وزارة الثقافة بالجزائر يستلمها من يصر فالمال العام بغير وجه حق.
2	Positif fort	Félicitation à nous tous, vous avez fait le bon choix. La journaliste leila bouzidi est réellement compétente et d'une personnalité professionnelle lui permettant de bien gérer sa carrière que dieu soit avec elle.	مبروك علينا وعليكم لقد أحسنت الاختيار . الصحفية القديرة ليلى بوزيدي حقا هي متمكنة وذات شخصية مهنية متحكمة في إدارة مهنتها أعانها الله و وفقها.
4	Négatif faible	Franchement, c'est Article non professionnel de la part d'un expert connu voulant la réussite de l'équipe national.	مقال غير احترافي صراحة من إعلامي كبير يحرص على نجاح المنتخب و قاسي كذلك.
6	Négatif fort	Franchement, je n'ai pas aimé le style de l'écrivain, et aussi son point de vus envers ce sujet et j'ai haï ses moqueries pour l'art.	صراحة لم أحب أبدا أسلوب الكاتب و لا رأيه في ا لموضوع و كرهت استهزاءه بالفن.
8	Neutre	Non, c'est son point de vue personnel	كلا انه رأيه الشخصي.

TABLE 1 – Exemple de commentaires

5.2 Extraction des caractéristiques

Nous commençons la phase d'extraction de caractéristiques par la construction manuelle des tables des marqueurs d'opinions du corpus utilisé définies comme suit :

- **Marqueur :**
  La table marqueur contient tous les prédicats, les adjectifs et les adverbes construisent à partir du corpus avec leurs polarités et intensités.
  Exemples:
  - **Prédicat:** أحبaimer, كرهdétester, ظنpenser.
  - **Adjectif:**جميلةbien fait, رائعةmagnifique, ركيكةlâche.
  - **Adverbe:**غنيةriche, مضجرةfatigante, مفيدةintéressante.
- **Intensité:**كثيراbeaucoup,جداtrès, مئةبالمئةcent pour cent.
- **Négation:**لا (non, ni, pas), لن لم ليسpas.
- **Adressage:**أنتtu, يا سيدي الكاتبMr l'écrivain.
- **Réflexivité:**أنا moi, رأيي mon opinion.

5.3 Les caractéristiques de représentation d'opinion

Nous avons aussi adoptés un ensemble des caractéristiques inspirées de celle des travaux de (Boughanem et al, 2010) et qui ont montrés leur efficacité dans la représentation d'un commentaire.

➢ **Adverbe**

Total-adverbes : tot(adv) = Nombre total des adverbes du document

Moy-adverbes = tot(adv)/(tot(adj) + tot(adv) + tot(pred)(1)

➢ **Adjectif**

Total-adjectives : tot(adj) = Nombre total des adjectifs du document

Moy-adjectives= tot(adj)/(tot(adj) + tot(adv) + tot(pred)(2)

➢ **Émotivité**

Les chercheurs ont exploité la présence des adverbes et des adjectifs dans un document comme un indicateur qui permet de déterminer les opinions. Nous calculons l'émotivité d'un document en comptant le nombre des adverbes et des adjectifs dans ce document.

$$Emot(d) = \frac{|\{\omega \in d \setminus type(\omega) \in \{adjectif, adverbe\}\}|}{|\{\omega \in d \setminus type(\omega) \in \{predicat\}\}|} \qquad (3)$$

➢ **Adressage**

La plupart des phrases trouvées dans les blogs et les forums contiennent des mots comme suit «أنتtoi, vousانتم, toi-mêmeنفسك, vous-mêmeأنفسكم, ilهو, elleهي, ilsهم, lui-mêmeنفسه, elle-mêmeنفسها» car les utilisateurs écrivent des commentaires sur un sujet, en s'adressant aux autres. De ce fait l'utilisation de ces pronoms d'adressage est très fréquente. Par conséquent, nous considérons que la composante d'adressage dans le cadre de notre détection d'opinions, est comme suit :

$$Add(d) = \frac{|\{\omega \cap \omega' \setminus \omega \in d, \omega' \in A\}|}{|A|+|R|} \qquad (4)$$

- $|\{\omega \cap \omega' \backslash \omega \epsilon d, \omega' \epsilon A\}|$ Représente le nombre d'occurrences des termes d'adressages dans le document *d* qui appartiennent à la liste d'adressage |A| que nous avons préparé.
- |A| est égal au nombre total de pronoms dans la liste d'adressage A.
- |R| est le nombre total de pronoms dans la liste de réflexivité.

➢ **Réflexivité**

Les blogueurs utilisent beaucoup de pronoms réflexifs comme « أنا شخصيا, أنا » « moi, moi-même» lors de l'écriture. Par exemple, l'utilisation de «ي» dans « رأيي »« Je pense que », « من وجهة نظري », « mon point de vu est que », etc. Toutes ces phrases font référence àune opinion d'opinion, et par conséquent, nous incluons la mesure de la réflexivité. L'idée est que tout document avec un plus grand nombre de ces mots sera plus subjectif par rapport à celui qui a moins de nombre de ces mots. Cette mesure est exprimée par la réflexivité *Ref (d)*.

$$Ref(d) = \frac{|\{\omega \cap \omega' \backslash \omega \epsilon d, \omega' \epsilon R\}|}{|R|+|A|} \qquad (5)$$

- $|\{\omega \cap \omega' \backslash \omega \epsilon d, \omega' \epsilon R\}|$ est le nombre de pronoms réfléchis dans le document *d* qui appartiennent à la liste de réflexivité R que nous avons construit.
- | R | est le nombre total de pronoms dans la liste de réflexivité R.
- |A| est l'ensemble de nombre total de pronoms dans la liste d'adressage.

En plus il y a un ensemble de caractéristiques que nous avons proposées qui sont décrites dans le tableau suivant :

Phrase	Nombre de phrases	∑ phrases
Positif fort	Nombre des mots positifs forts	∑ mot (posFo)
Positif faible	Nombre des mots positifs faibles	∑ mot (posFa)
Négatif fort	Nombre de mots négatifs forts	∑ mot (negFo)
Négatif faible	Nombre de mots négatifs faibles	∑ mot (negFa)
Neutre	Nombre de mots neutres	∑ mot (ntr)
Prédicat	Total-prédicats	Nbr(pred)=Nombre total des prédicats du document
	Moy- prédicats	Nbr(pred)/(tot(adj) + tot(adv) + tot(pred)
Polarité	Somme polarité	SomPolarite= ∑mot(posFo) + ∑mot(posFa) + ∑ mot (negFo) + ∑ mot(negFa) + ∑ mot (ntr)
	Mots positifs forts	∑ mot(posFo)/SomPolarite
	Mots positifs faibles	∑ mot(posFa)/SomPolarite
	Mots négatifs forts	∑ mot(negFo)/SomPolarite
	Mots négatifs faibles	∑ mot(negFa)/SomPolarite
	Mots neutres	∑ mot(ntr)/SomPolarite

TABLE 2 – Les mesures proposées

5.4 La classification

Ils existent plusieurs méthodes de classification supervisée et beaucoup d'entre elles ont été testées pour la classification d'opinions. On peut citer les arbres de décision, les réseaux de neurones, la régression logistique, les règles de décision ainsi que des méthodes combinant différents classifieurs comme les systèmes de votes ou les algorithmes de Boosting. Toutefois, les méthodes les plus présentes dans la littérature, et qui semblent également être les plus performantes sur les textes, sont les machines à support de vecteurs (Pang et Lee, 2004),(Wilson et al, 2004),(Nigam et Hurst, 2006),(Généreux et Santini, 2007), (Trinh, 2007),(Crestan et al, 2007),(Plantié et al, 2008),(Kaiser et al, 2010), (Poirier et al, 2011),(Abbasi et al, 2008),(Montassir et al, 2012).

Au premier temps, les SVM sont conçus pour traiter le problème de classification binaire. Leur extension aux problèmes de classification multi- classes demeure un sujet de recherche très actif. En général, les méthodes multi-classes proposées dans littérature suivent l'une des deux approches suivantes :

- ✓ La combinaison de plusieurs classifieurs binaires classiques.
- ✓ La conception d'un seul classifieur SVM en résolvant un seul problème d'optimisation.

Dans notre travail nous avons adopté ici les deux stratégies de la première approche « un-contre-tous » et « un contre un » pour décider laquelle entre ces deux est meilleure pour la classification d'opinions multi classes.

5.5 La combinaison des classifieurs

Plutôt que de chercher à optimiser un seul classifieur en choisissant les meilleures caractéristiques pour un problème donné, les chercheurs ont trouvé plus intéressant de combiner plusieurs méthodes de classification.

La multiplication des travaux sur la combinaison de classifieurs a entraîné la mise au point de nombreux schémas traitant les données de manières différentes. Nous utilisons dans cette partie la combinaison parallèle. L'approche parallèle laisse les différents classifieurs opérer indépendamment les uns des autres puis fusionne leurs réponses respectives.

Notre objectif est d'analyser le comportement d'un système combinant plusieurs classifieurs afin d'augmenter les performances de classification. Donc, pour construire un système multi classifieurs, il y a une méthode qui se base sur l'utilisation d'un même classifieur en modifiant à chaque fois ses paramètres internes ; ce qui va générer des classifieurs différents. Cette différence sera traduite par une complémentarité durant le processus de classification d'un commentaire inconnu.

En effet, Vu que chaque classifieurs SVM selon la fonction noyau génère un résultat différent, et vu que dans la littérature, on n'a pas pu prouver la supériorité d'une fonction par rapport aux autres dans le cas général, on a utilisé quatre classifieurs, chacun d'eux à une fonction noyau différente qui sont les suivantes (linéaire, polynomiale, gaussienne et tangente). En combinant ces classifieurs, chacun d'eux va assurer un certain niveau de complémentarité au système global.

Le système généré est construit de 4 classifieurs, où chacun d'eux offre 5 sorties. Afin de fusionner les résultats des 4 classifieurs, nous avons appliqué deux méthodes de

combinaison les plus connus (le vote majoritaire et le vote pondéré).

6 L'évaluation du système

Au niveau du test on applique notre système de classification sur un nouvel ensemble différent de celui de l'apprentissage, contient 30 commentaires étiquetés manuellement.

Pour évaluer les résultats obtenus il faut calculer le taux de chaque classe pour chaque stratégie (un contre tous et un contre un) et pour chaque méthode de fusion (le vote et le vote pondéré).

6.1 La stratégie un contre tous

	Linéaire	Polynomiale	Gaussienne	Tangente	Vote	Vote pondéré
Neutre	0.19	0.27	0.31	0.09	0.30	0.39
Positif fort	0.69	0.75	**0.82**	0.20	**0.81**	**0.91**
Positif faible	0.40	0.49	0.55	0.11	0.55	0.59
Négatif fort	**0.89**	**0.90**	**1.00**	0.31	**1.00**	**1.00**
Négatif faible	0.45	0.51	0.59	0.20	0.59	0.60

TABLE 3 – Variation du taux de classification de chaque classe pour les quatre fonctions noyaux et avec les deux méthodes de fusion

A partir du tableau ci-dessus on peut constater que les classifieurs SVMs conçus avec les fonctions noyaux polynomiale et gaussienne donne le meilleur taux de classification par comparaison aux autres SVMs.

On constate notamment que la classe de type « négatif fort » offre le meilleur taux de classification ; en revanche la classe neutre est celle ayant le taux le plus faible. Cela est dû à notre avis aux poids des marqueurs représentant ces deux classes.

On remarque que les résultats de la combinaison (soit pour la méthode de vote ou de vote pondéré) sont meilleurs qu'en utilisant un seul classifieur; en effet, malgré que la fonction tangente génère un taux de reconnaissance très faible mais sa présence dans le processus de combinaison enrichie la classification; d'où l'intérêt majeur de la combinaison de classifieurs.

Concernant les méthodes de combinaison, on a testé les deux méthodes les plus utilisées pour les classifieurs de type classe qui sont « vote majoritaire et vote pondéré» afin de maintenir celle qui offre le taux de classification le plus fort. Dans notre application, on constate que le taux du vote pondéré est supérieur de celui de vote majoritaire.

6.2 La stratégie un contre un

	Linéaire	Polynomiale	Gaussienne	Tangente	Vote	Vote pondéré
Neutre	0.50	0.57	0.61	0.29	0.62	0.65
Positif fort	**0.81**	**0.91**	**1.00**	0.31	**1.00**	**1.00**
Positif faible	**0.88**	**0.88**	**0.88**	**1.00**	**0.88**	**0.95**
Négatif fort	**0.88**	**0.98**	**1.00**	0.23	**1.00**	**1.00**
Négatif faible	0.49	0.50	0.62	0.12	0.62	0.68

TABLE 4 – Variation du taux de classification de chaque classe pour les quatre fonctions noyaux et avec les deux méthodes de fusion

D'après le tableau ci dessus on constate que la fonction gaussienne donne de bon taux de classification pour les quatre classes (neutre, positif faible, négatif fort et négatif faible), par contre la fonction tangente donne le meilleur taux pour la classe positif faible.

En générale les deux méthodes de fusion produisent des bons résultats, mais c'est la méthode de vote pondéré qui a le meilleur taux pour toutes les classes.

Nous concluons alors, que pour améliorer le taux de classification, il est préférable d'utiliser la méthode de vote pondéré.

Après la mise en œuvre des stratégies, il s'avère que les résultats obtenus par la stratégie « un contre un » sont meilleurs que ceux de la stratégie « un contre tous ». En effet comme le montre le tableau précédant (Table 4), le taux de toutes les classes est augmenté pour tous les fonctions, ce qui signifie que la classification est excellente.

7 Conclusion

Au terme de ce travail, nous procédons dans les lignes qui se suivent à un récapitulatif du travail effectué. Rappelons que notre travail consistait essentiellement à développer une application qui permet la détection de polarité d'opinions dans les forums en langue arabe par combinaison de plusieurs SVM. Ce système a pour rôle d'extraire les caractéristiques représentant les commentaires du corpus et de les classifier en catégories par la coopération de plusieurs classifieurs SVMs.

Donc notre système opère en trois phases, la première consiste à la construction et le prétraitement manuel du corpus recueillis à partir des journaux arabes algériens. La seconde phase est une extraction des caractéristiques afin de détecter et représenter les commentaires. Le choix de ces caractéristiques est fait par une recherche approfondie sur les plus importantes caractéristiques pouvant représenter de mieux le commentaire tout en évitant la redondance et la confusion des données d'entrée, et nous avons conclu qu'il est difficile de faire le choix des bons caractéristiques. Enfin la troisième phase est la réalisation du module de classification et dans le but de bénéficier des avantages des systèmes multi-classifieurs, on a proposé un système combinant 04 classifieurs SVMs multi-classes représentant chacun par une fonction noyau différente pour les deux stratégies (un contre tous et un contre un). Les résultats issus de la combinaison sont très encourageants et nous ont permis de mieux s'investir dans cet axe tout en analysant le rôle que peut jouer chaque paramètre des caractéristiques.

Les tests sur les commentaires en langue arabe avec les deux stratégies « un contre tous » et « un contre un », nous ont permis de prouver que la stratégie un contre un donne de meilleurs résultats avec les commentaires des journaux en langue arabe.

Cette expérience, nous a permis de faire une première exploration du vaste domaine qui est l'opinion mining, et nous incite à aller plus loin dans ce domaine dans le cadre de travaux futurs, car l'opinion mining représente un axe de recherche très prometteur et très passionnant, ce qui explique l'intérêt recrudescent que portent les chercheurs pour ce domaine.

Comme perspective par rapport à ce travail, il serait intéressant de généraliser notre système avec n'importe quelle base de commentaires.

Une autre perspective consiste à utiliser d'autres caractéristiques afin d'enrichir le vecteur des caractéristiques pour une meilleure représentation des commentaires, et optez pour l'ajout d'un module de sélection de caractéristiques.

Il serait aussi intéressant de combiner plusieurs algorithmes de classification tels que les SVM (machine à vecteur de support) et les réseaux bayesiens, pour pouvoir évaluer et améliorer les performances.

Références

ABBASI, A., CHEN, H. et SALEM, A. (2008*).* Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums.*ACM Transactions on Information Systems Volume 26 Issue 3, Jun. 2008.*

ALMAS, Y. et AHMAD, K. (2007). A Note on Extracting 'Sentiments' in Financial News in English, Arabic & Urdu.*In Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages Linguistic Institute, Stanford, California, USA.*July 21-22, 2007.

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V. et Jurafsky, D. (2004).Automatic extraction of opinion propositions and their holders. *Actes d' AAAI'04, 2004.*

BOSCA, A. et DINI, L. (2009).User Logs as a Means to Enrich and Refine Translation Dictionaries.Workshop of Cross-Language Evaluation Forum.*CLEF (1) 2009.*

BOUGHANEM, M. et BELBACHIR, F. (2010*).*Expérimentation de fonctions pour la détection d'opinions dans les blogs (mémoire de Master).

CHKLOVSKI, T. (2006). Deriving quantitative overviews of free text assessments on the web.*Actes d' IUI'06.*

CRESTAN, E. et ACUNA-AGOST,R. (2007).Quel modèle pour détecter une opinion ? Trois propositions pour généraliser l'extraction d'une idée dans un corpus. *In Actes de l'atelier de clôture du 3ème défi fouille de textes , Grenoble, France.*

DINI, L. (2002).Compréhension multilingue et extraction de l'information. *In , F. Segond (ed.), Multilinguisme et traitement de l'information (Traité des sciences et techniques de l'information), Editions Hermes Science, 2002.*

DINI, L. etMAZZINI, G. (2002).Opinion classification through information extraction.*In A. Zanasi, C. A. Brebbia, N. F. F. Ebecken, P. Melli (Eds), Data Mining III, WIT Press, 2002.*

EL-HALEES, A. (2011). Arabic Opinion Mining Using Combined Classification Approach.*In Proceeding The International Arab Conference On Information Technology, Azrqa, Jordan 2011.*

ELHAWARY, M. et ELFEKY, M. (2010).Mining Arabic Business Reviews. *IEEE International Conference on Data Mining Workshops. 2010.*

GÉNÉREUX, M. et SANTINI, M.(2007).Défi : Classification de textes français subjectifs. *Actes de l'atelier de clôture du 3e DÉfi Fouille de Textes, AFIA, Grenoble, France, 2007.*

HATZIVASSILOGLOU, V. et MCKEOWN, K. R. (1997).Predicting the semantic orientation of adjectives. *Actes d' ACL'97, 1997.*

KIM, S.-M. et HOVY, E. (2004).Determining the sentiment of opinions. *Actes de COLING'04, 2004.*

KAISER, C., KROCKEL, J. et BODENDORF, F. (2010). Swarm Intelligence for Analyzing Opinions in Online Communities.

LIU, B. (2010).Sentiment Analysis. *Invited talk at the 5th Annual Text Analytics Summit.*

MAUREL, S., CURTONI, P. et DINI, L.(2007). L'analyse des sentiments dans les forums.

MAUREL, S., CURTONI, P.et DINI, L. (2008).L'analyse des sentiments dans les forums.*Actes de l'atelier FOuille des Données d'OPinions, 2008.*

MILGRAM, J. (2007).Contribution à l'intégration des machines à vecteurs de support au sein des systèmes de reconnaissance de formes: application à la lecture automatique de l'écriture manuscri.

Mountassir, A., Benbrahim, H. et Berrada, I. (2012). A cross-study of Sentiment Classification on Arabic corpora.*In Research and Development in Intelligent Systems XXIX. Springer London, 2012.*

NIGAM, K. ET HURST, M. (2006). Towards a Robust Metric of Polarity, Computing Attitude and Affect in Text: Theory and Applications. *Springer, Dordrecht, The Netherlands, 2006.*

PANGB, et Lee L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.

PANG, B.etLEE, L. (2008).Opinion Mining and Sentiment Analysis.

PLANTIÉ, M., ROCHE, M., DRAY, G. et PONCELET, P. (2008).Is a Voting Approach Accurate for Opinion Mining ?.

POIRIER, D., FESSANT, F. et TELLIER, I. (2011).De la classification d'opinions à la recommandation : l'apport des textes communautaires.*In TALN 2011 (Traitement automatique des langues naturelles).*

RUSHDI, S., MOHAMMED, M., MARTÍN-VALDIVIA, T., UREÑA-LÓPEZ, A. L., et JOSÉ M. (2011).OCA: Opinion corpus for Arabic. *Journal of the American Society for Information Science and Technology 62, no. 10 (2011): 2045-2054.*

TRINH, A.-P. (2007). Classification de texte et estimation probabiliste par machine à vecteur de support. *In Actes de l'atelier de clôture du 3ème défi fouille de textes , Grenoble, France.*

TURNEY, P. (2002).Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the Association for Computational Linguistics, ACL'02, 2002.*

WILSON, T. WIEBE, J. et HWA, R. (2004). Just How Mad Are You? Finding Strong and Weak Opinion Clauses.

YU, H. et HATZIVASSILOGLOU, V. (2003).Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *Actes d' EMNLP'03, 2003.*

État de l'art des méthodes d'extraction automatique de termes-clés

Adrien Bougouin
LINA - UMR CNRS 6241, Université de Nantes, France
adrien.bougouin@univ-nantes.fr

RÉSUMÉ

Cet article présente les principales méthodes d'extraction automatique de termes-clés. La tâche d'extraction automatique de termes-clés consiste à analyser un document pour en extraire les expressions (phrasèmes) les plus représentatives de celui-ci. Les méthodes d'extraction automatique de termes-clés sont réparties en deux catégories : les méthodes supervisées et les méthodes non supervisées. Les méthodes supervisées réduisent la tâche d'extraction de termes-clés à une tâche de classification binaire (tous les phrasèmes sont classés parmi les termes-clés ou les non termes-clés). Cette classification est possible grâce à une phase préliminaire d'apprentissage, phase qui n'est pas requise par les méthodes non-supervisées. Ces dernières utilisent des caractéristiques (traits) extraites du document analysé (et parfois d'une collection de documents de références) pour vérifier des propriétés permettant d'identifier ses termes-clés.

ABSTRACT

State of the Art of Automatic Keyphrase Extraction Methods

This article presents the state of the art of the automatic keyphrase extraction methods. The aim of the automatic keyphrase extraction task is to extract the most representative terms of a document. Automatic keyphrase extraction methods can be divided into two categories : supervised methods and unsupervised methods. For supervised methods, the task is reduced to a binary classification where terms are classified as keyphrases or non keyphrases. This classification requires a learning step which is not required by unsupervised methods. The unsupervised methods use features extracted from the analysed document (sometimes a document collection) to check properties which allow keyphrase identification.

MOTS-CLÉS : extraction de termes-clés ; méthodes supervisées ; méthodes non-supervisées ; état de l'art .

KEYWORDS: keyphrase extraction ; supervised methods ; unsupervised methods ; state of the art .

1 Introduction

Les termes-clés sont des mots ou des expressions (multi-mots) représentant les aspects principaux qui sont abordés dans un document. De ce fait, ils sont utilisés dans de nombreux domaines du Traitement Automatique des Langues (TAL). Turney (1999) émet l'hypothèse qu'ils peuvent faciliter la lecture d'un utilisateur en lui permettant de surfer d'un point clé à un autre lorsqu'ils

sont mis en évidence dans un texte. D'autres chercheurs utilisent leurs vertus synthétiques dans des méthodes de construction automatique de résumés (Wan *et al.*, 2007; Litvak et Last, 2008; Boudin et Morin, 2013), mais ils s'avèrent surtout de plus en plus utiles avec l'essor de l'Internet et la disponibilité de nombreux documents numériques qu'il faut pouvoir indexer de manière pertinente pour faciliter leur recherche par des utilisateurs (Medelyan et Witten, 2008). Dans ce contexte de recherche d'information, les termes-clés peuvent aussi être directement bénéfiques aux utilisateurs en servant de suggestions à une requête qu'ils essaient de formuler (Jones et Staveley, 1999).

Bien que les termes-clés soient utiles pour de multiples tâches, très peu de documents en sont pourvus, du fait du coût important de production de ceux-ci, en termes de temps et de ressources humaines. Pour y remédier de nombreux chercheurs s'intéressent à l'extraction automatique de ceux-ci et certaines campagnes d'évaluations, telles que DEFT (Paroubek *et al.*, 2012) et SemEval (Kim *et al.*, 2010), proposent des tâches d'extraction automatique de termes-clés dans le but de confronter les différents systèmes existants. Pour ce faire, les données et la méthode d'évaluation sont les mêmes pour tous les systèmes.

Il existe aussi une autre tâche nommée assignation automatique de termes-clés. Cette tâche est très proche de l'extraction automatique de termes-clés, mais elle est plus contrôlée. Elle consiste aussi à donner un ensemble de termes-clés pour un document, mais certains de ces termes peuvent ne pas être présents dans celui-ci. Ceci est dû au fait que les méthodes d'assignation de termes-clés utilisent des ressources supplémentaires telles que des référentiels terminologiques. Ceux-ci contiennent des termes spécifiques au(x) domaine(s) traité(s) et l'assignation de ces termes peut être déclenchée par la présence de certains autres dans le document analysé.

Dans cet article, seules les méthodes d'extraction automatique de termes-clés sont présentées. Celles-ci appartiennent à deux catégories distinctes : les méthodes supervisées et les méthodes non-supervisées. Dans le cas supervisé, l'extraction des termes-clés est effectuée grâce à un apprentissage préalable servant à calibrer la méthode avec un corpus dont les documents sont annotés en termes-clés. Les méthodes non-supervisées ne requièrent pas de phase d'apprentissage. Elles exploitent des représentations efficaces des documents ainsi que des propriétés définies à partir de traits statistiques pour extraire les termes-clés parmi des termes candidats.

Dans la section 2 de cet article, nous présentons les méthodes existantes d'extraction automatique de termes-clés, en commençant par les méthodes non-supervisées, puis les méthodes supervisées. Dans la section 3 nous terminons par un bilan de l'état de l'art et nous discutons des perspectives de travaux futurs.

2 Les méthodes d'extraction automatique de termes-clés

L'extraction de termes-clés est une tâche qui consiste à analyser un document et à en extraire les aspects importants. Alors que les méthodes de résumé automatique utilisent des phrases pour construire une vision synthétique du document, l'extraction de termes-clés se focalise sur les unités textuelles qui composent ces phrases. Un ensemble de termes-clés peut donc être perçu comme un résumé dont les points clés sont exprimés sans liaisons entre eux. Les unités textuelles sur lesquelles travaillent les systèmes d'extraction automatique de termes-clés sont appelées termes candidats. Ces derniers sont des mots ou des multi-mots (phrasèmes) pouvant

être promus au statut de terme-clé.

L'extraction de termes candidats est une étape préliminaire de l'extraction de termes-clés, que ce soit pour les méthodes non-supervisées ou supervisées. Cette étape est importante, car si certains termes-clés du document analysé ne sont pas présents dans l'ensemble des termes candidats, alors ceux-ci ne pourront pas être extrait. Hulth (2003) étudie trois méthodes d'extraction des termes candidats. L'une consiste à extraire les chunks nominaux [1], tandis que les deux autres extraient tous les n-grammes et les filtrent, soit pour retirer les termes contenant des mots outils dans le premier cas, soit pour ne retenir que les termes respectant certains patrons syntaxiques dans le second cas (usage des parties du discours). Dans ses expériences Hulth (2003) montre que l'extraction de termes-clés à partir de n-grammes filtrés avec les mots outils donne les meilleurs résultats parmi les trois méthodes qu'elle propose.

Les travaux de Hulth (2003) sont évalués avec un corpus dont les documents sont des résumés d'articles scientifiques. Cependant, dans d'autres domaines tels que la bio-médecine, la nature des termes à extraire n'est pas la même. En effet, ce sont les acronymes et les entités nommées (noms de protéines par exemple) qu'il est nécessaire d'extraire en tant que termes-clés (Nobata *et al.*, 2008). Pour cela, l'extraction de termes candidats est spécifique au domaine d'application. Les méthodes d'extraction de termes-clés présentées dans cet article traitent des documents supposés sans spécificités particulières, les méthodes d'extraction de termes candidats sont donc les mêmes que celles expérimentées par Hulth (2003), mais il est envisageable de les adapter à des domaines présentant des spécificités particulières.

Utilisés avec les méthodes non-supervisées, les termes candidats sont ordonnés selon un score d'importance obtenu soit à partir d'eux-mêmes, soit à partir de l'importance des mots qui les composent. Si une méthode s'appuie uniquement sur les mots, alors le score d'un terme candidat est généralement calculé en faisant la somme des mots qui le composent. Cependant, ceci n'est pas toujours juste, c'est donc un inconvénient important des méthodes travaillant sur les mots pour extraire les termes-clés. En effet, la sommation peut privilégier des termes qui contiennent beaucoup de mots non-importants vis-à-vis de termes contenant très peu de mots, mais importants.

Utilisés dans les méthodes supervisées, les termes candidats sont classés en tant que termes-clés ou non termes-clés grâce à des méthodes de classification.

2.1 Méthodes non-supervisées

Les méthodes non-supervisées d'extraction de termes-clés ont la particularité de s'abstraire du domaine et de la langue des documents à analyser [2]. Cette abstraction est due au fait que les termes candidats sont analysés avec des règles simples déduites à partir de traits statistiques issus seulement du texte analysé, ou bien d'un corpus de référence non annoté.

De nombreuses approches sont proposées. Certaines se fondent uniquement sur des statistiques alors que d'autres les combinent avec des représentations plus complexes des documents. Ces

1. Un chunk est une unité minimale de sens constituée d'un ou de plusieurs mots. Un chunk nominal est un chunk dont la tête est un nom ou un pronom. Par exemple, dans « Nous avons une bonne politique qualitative. », « Nous » et « une bonne politique qualitative » sont des chunks nominaux.

2. L'abstraction de la langue est vraie pour ce qui est de la méthodologie, cependant les pré-traitements tels que la segmentation en phrases, en mots et l'étiquetage en parties du discours sont eux spécifiques à la langue.

représentations peuvent aller de groupes de mots sémantiquement similaires à des graphes dont les nœuds sont des unités textuelles (mots, expressions, phrases, etc.) liées par des relations de recommandation [3].

2.1.1 Approches statistiques

Plusieurs approches cherchent à définir ce qu'est un terme-clé en s'appuyant sur certains traits statistiques et en étudiant leur rapport avec la notion d'importance d'un terme candidat. Plus un terme candidat est jugé important vis-à-vis du document analysé, plus celui-ci est pertinent en tant que terme-clé.

TF-IDF (cf. équation 1) de Jones (1972) et Likey (cf. équation 2) de Paukkeri et Honkela (2010) sont deux méthodes qui comparent le comportement d'un terme candidat dans le document analysé avec son comportement dans une collection de documents (corpus de référence). L'objectif est de trouver les termes candidats dont le comportement dans le document varie positivement comparé à leur comportement global dans la collection. Dans les deux méthodes ceci s'exprime par le fait qu'un terme a une forte importance vis-à-vis du document analysé s'il y est très présent, alors qu'il ne l'est pas dans le reste de la collection.

$$\text{TF-IDF}(\text{terme}) = TF(\text{terme}) \times \log\left(\frac{N}{DF(\text{terme})}\right) \tag{1}$$

$$Likey(\text{terme}) = \frac{\text{rang}_{\text{document}}(\text{terme})}{\text{rang}_{\text{corpus}}(\text{terme})} \tag{2}$$

Dans TF-IDF, TF représente le nombre d'occurrences d'un terme dans le document analysé et DF représente le nombre de documents dans lequel il est présent, N étant le nombre total de documents. Plus le score TF-IDF d'un terme candidat est élevé, plus celui-ci est important dans le document analysé. Dans Likey, le rang d'un terme candidat dans le document et dans le corpus est obtenu à partir de son nombre d'occurrences, respectivement dans le document et dans le corpus de référence. Plus le rapport entre ces deux rangs est faible, plus le terme candidat évalué est important dans le document analysé.

Okapi (ou BM25) (Robertson *et al.*, 1999) est une mesure alternative à TF-IDF. En Recherche d'Information (RI), celle-ci est plus utilisée que le TF-IDF. Bien que l'extraction automatique de termes-clés soit une discipline à la frontière entre le TAL et la RI, la méthode de pondération Okapi n'a, à notre connaissance, pas été appliquée pour l'extraction de termes-clés. Dans l'article de Claveau (2012), Okapi est décrit comme un TF-IDF prenant mieux en compte la longueur des documents. Cette dernière est utilisée pour normaliser le TF (qui devient TF_{BM25}) :

$$\text{Okapi}(\text{terme}) = TF_{BM25}(\text{terme}) \times \log\left(\frac{N - DF(\text{terme}) + 0,5}{DF(\text{terme}) + 0,5}\right) \tag{3}$$

$$TF_{BM25} = \frac{TF(\text{terme}) \times (k_1 + 1)}{TF(\text{terme}) + k_1 \times \left(1 - b + b \times \frac{DL}{DL_{\text{moyenne}}}\right)} \tag{4}$$

3. Pour une étude comparative de certaines des méthodes par regroupement (Liu *et al.*, 2009) et à base de graphe (Mihalcea et Tarau, 2004; Wan et Xiao, 2008b), voir l'article de Hasan et Ng (2010).

Dans la formule (4), k_1 et b sont des constantes fixées à 2 et 0,75 respectivement. DL représente la longueur du document analysé et $DL_{moyenne}$ la longueur moyenne des documents de la collection utilisée.

Barker et Cornacchia (2000) estiment que les grands phrasèmes sont plus informatifs et qu'ils doivent être privilégiés. Pour cela, leur approche est très simple : plus un groupe nominal est long et fréquent dans le document analysé, plus il est jugé pertinent en tant que terme-clé de ce document. Cependant, pour éviter la répétition dans le texte, les auteurs des documents utilisent les même expression sous des formes alternatives (plus courtes, par exemple). La fréquence d'une expression ne reflète donc pas forcément sa fréquence réelle d'utilisation, car celle-ci est répartie dans les différentes alternatives. De ce fait, Barker et Cornacchia (2000) repèrent dans les groupes nominaux la tête nominale et utilisent en plus la fréquence de celle-ci.

Tomokiyo et Hurst (2003) tentent de vérifier deux propriétés, en utilisant des modèles de langue uni-grammes et n-grammes et en calculant leur divergence (Kullback-Leibler). Les deux propriétés qu'ils tentent de vérifier sont les suivantes :

- La grammaticalité : un terme-clé doit être bien formé syntaxiquement.
- L'informativité : un terme-clé doit capturer au moins une des idées essentielles exprimées dans le document analysé.

Pour un terme candidat donné, plus sa probabilité en passant du modèle uni-gramme généré à partir du document vers le modèle n-gramme généré à partir du même document augmente, plus il respecte la propriété de grammaticalité. De même, plus sa probabilité en passant du modèle n-gramme généré à partir d'un corpus de référence vers le modèle n-gramme généré à partir du document analysé augmente, plus le terme candidat est informatif.

La méthode que propose Ding *et al.* (2011) utilise TF-IDF comme indicateur de l'importance d'un terme-clé. Dans un ensemble, cette importance doit être maximisée pour chaque terme-clé, mais les auteurs estiment que ceci n'est pas suffisant. Comme Tomokiyo et Hurst (2003), ils définissent deux propriétés qui doivent être respectées :

- La couverture : un ensemble de termes-clés doit couvrir l'intégralité des sujets abordés dans le document représenté.
- La cohérence : les termes-clés doivent être cohérents entre eux.

La propriété de couverture est évaluée avec le modèle *Latent Dirichlet Allocation* (LDA) qui donne la probabilité d'un terme candidat sachant un sujet. La cohérence est évaluée pour chaque paire de termes-clés de l'ensemble avec la mesure d'information mutuelle. Ces deux propriétés sont définies comme contraintes que les auteurs utilisent avec une méthode de programmation par les entiers (technique d'optimisation), la maximisation de la pertinence de chaque terme-clé étant l'objectif à atteindre.

Les traits statistiques utilisés dans les méthodes précédentes sont uniquement utilisés pour déterminer un score de pertinence des termes candidats en tant que termes-clés. Une donnée statistique non citée précédemment, mais pourtant récurrente dans les méthodes d'extraction de termes-clés, est la fréquence de co-occurrences entre deux phrasèmes (termes). Deux phrasèmes co-occurrent s'ils apparaissent ensemble dans le même contexte. La co-occurrence peut être calculée de manière stricte (les phrasèmes doivent être côte-à-côte) ou bien dans une fenêtre de mots. Compter le nombre de co-occurrences entre deux termes permet d'estimer s'ils sont sémantiquement liés ou non. Ce lien sémantique à lui seul ne peut pas servir à extraire des

termes-clés, mais il permet de mieux organiser les termes d'un document pour affiner l'extraction (Matsuo et Ishizuka, 2004; Liu *et al.*, 2009; Mihalcea et Tarau, 2004).

2.1.2 Approches par regroupement

L'objectif des approches par regroupement est de définir des groupes dont les unités textuelles partagent une ou plusieurs caractéristiques communes. Ainsi, lorsque des termes-clés sont extraits à partir de chaque groupe, cela permet de mieux couvrir le document analysé selon les caractéristiques utilisées.

Dans la méthode de Matsuo et Ishizuka (2004), ce sont les termes (phrasèmes) qui sont regroupés. Parmi ceux-ci, seuls les plus fréquents sont concernés par le regroupement. Celui-ci s'effectue en fonction du lien sémantique [4] entre les termes. Après le regroupement, la méthode consiste à comparer les termes candidats du document analysé avec les groupes de termes fréquents, en faisant l'hypothèse qu'un terme candidat qui co-occurre plus que selon toute probabilité avec les termes fréquents d'un ou plusieurs groupes est plus vraisemblablement un terme-clé.

Dans l'algorithme KeyCluster, Liu *et al.* (2009) utilisent aussi un regroupement sémantique, mais dans leur cas ils considèrent les mots du document analysé et ils excluent les mots outils. Dans chaque groupe sémantique, le mot qui est le plus proche du centroïde est sélectionné comme mot de référence. L'ensemble des mots de référence est ensuite utilisé pour filtrer les termes candidats en ne considérant comme termes-clés que ceux qui contiennent au moins un mot de référence (tous les mots de référence devant être utilisés dans au moins un terme-clé).

2.1.3 Approches à base de graphe

Les approches à base de graphe consistent à représenter le contenu d'un document sous la forme d'un graphe. La méthodologie appliquée est issue de PageRank (Brin et Page, 1998), un algorithme d'ordonnancement de pages Web (nœuds du graphe) grâce aux liens de recommandation qui existent entre elles (arcs du graphe). TextRank (Mihalcea et Tarau, 2004) et SingleRank (Wan et Xiao, 2008b) sont les deux adaptations de base de PageRank pour l'extraction automatique de termes-clés [5]. Dans celles-ci, les pages Web sont remplacées par des unités textuelles dont la granularité est le mot et un arc est créé entre deux nœuds si les mots qu'ils représentent co-occurrent dans une fenêtre de mots donnée.

Le graphe est noté $G = (N, A)$, où N est l'ensemble des nœuds du graphe et où A est l'ensemble de ses arcs entrants et sortant : $A_{entrant} \cup A_{sortant}$ [6]. Pour chaque nœud du graphe, un score est calculé par un processus itératif destiné à simuler la notion de recommandation d'une unité textuelle par d'autres [7] (cf. équation 5). Ce score à chaque nœud n_i permet d'ordonner les mots par degré d'importance dans le document analysé. La liste ordonnée des mots peut ensuite être

4. Deux phrasèmes qui co-occurrent fréquemment ensemble sont jugés sémantiquement liés.
5. TextRank a aussi été utilisé pour faire du résumé automatique.
6. Dans le cas de TextRank et de SingleRank $A_{entrant} = A_{sortant}$, car le graphe n'est pas orienté.
7. Plus le score d'une unité textuelle est élevé, plus celle-ci est importante dans le document analysé.

utilisée pour extraire les termes-clés.

$$S(n_i) = (1-\lambda) + \lambda \times \sum_{n_j \in A_{\text{entrant}}(n_i)} \frac{p_{j,i} \times S(n_j)}{\sum_{n_k \in A_{\text{sortant}}(n_j)} p_{j,k}} \quad (5)$$

λ est un facteur d'atténuation qui peut être considéré ici comme la probabilité pour que le nœud n_i soit atteint par recommandation. $p_{j,i}$ représente le poids de l'arc allant du nœud n_j vers le nœud n_i, soit le nombre de co-occurrences entre les deux mots i et j [8].

Dans leurs travaux, Wan et Xiao (2008b) s'intéressent à l'ajout d'informations dans le graphe grâce à des documents similaires (voisins) et aux relations de co-occurrences qu'ils possèdent (ExpandRank). L'objectif est de faire mieux ressortir les mots importants du graphe en ajoutant de nouveaux liens de recommandation ou bien en renforçant ceux qui existent déjà. L'usage de documents similaires peut cependant ajouter ou renforcer des liens qui ne devraient pas l'être. Pour éviter cela, les auteurs réduisent l'impact des documents voisins en utilisant leur degré de similarité avec le document analysé. Une alternative à ExpandRank, CollabRank, également proposée par Wan et Xiao (2008a), fonctionne de la même manière, mais certains choix des auteurs rendent impossible l'usage du degré de similarité pour réduire l'impact des documents voisins. Les résultats moins concluants de CollabRank tendent à confirmer l'importance de l'usage du degré de similarité.

Dans l'optique d'améliorer encore TextRank/SingleRank, Liu *et al.* (2010) proposent une méthode qui cherche cette fois-ci à augmenter la couverture de l'ensemble des termes-clés extraits dans le document analysé (TopicalPageRank). Pour ce faire, ils tentent d'affiner le rang d'importance des mots dans le document en tenant compte de leur rang dans chaque sujet abordé. Le rang d'un mot pour un sujet est obtenu en intégrant à son score PageRank la probabilité qu'il appartienne au sujet (cf. équation 6). Le rang global d'un terme candidat est ensuite obtenu en fusionnant ses rangs pour chaque sujet.

$$S_{\text{sujet}}(N_i) = (1-\lambda) \times p(\text{sujet}|i) + \lambda \times \sum_{N_j \in A_{\text{entrant}}(N_i)} \frac{p_{j,i} \times S(N_j)}{\sum_{N_k \in A_{\text{sortant}}(N_j)} p_{j,k}} \quad (6)$$

Les approches à bases de graphe présentées ci-dessus effectuent toutes un ordonnancement des mots du document analysé selon leur importance dans celui-ci. Pour extraire les termes-clés il est donc nécessaire d'effectuer du travail supplémentaire à partir de la liste ordonnée de mots. Dans la méthode TextRank, les k mots les plus importants sont sélectionnés et retournés (après que ceux apparaissant en collocation dans le document aient été concaténés). La technique utilisée dans les autres méthodes consiste à ordonner les termes candidats en fonction de la somme du score des mots qui les composent. Cependant, puisque l'un des avantages du graphe est que les nœuds peuvent avoir une granularité contrôlée, Liang *et al.* (2009) décident d'utiliser des mots et des multi-mots au lieu de simples mots et de tirer profit de traits supplémentaires, la taille du terme ou encore sa première position dans le document analysé.

8. TextRank utilise un graphe non-pondéré. Dans ce cas, $p_{j,i}$ vaut toujours 1.

2.2 Méthodes supervisées

Les méthodes supervisées sont des méthodes capables d'apprendre à réaliser une tâche particulière, soit ici l'extraction de termes-clés. L'apprentissage se fait grâce à un corpus dont les documents sont annotés en termes-clés. L'annotation permet d'extraire les exemples et les contres-exemples dont les traits statistiques et/ou linguistiques servent à apprendre une classification binaire. La classification binaire consiste à indiquer si un terme candidat est un terme-clé ou non.

De nombreux algorithmes d'apprentissage sont utilisés dans divers domaines. Ils peuvent potentiellement s'adapter à n'importe quelle tâche, dont celle de l'extraction automatique de termes-clés. Les algorithmes utilisés pour celle-ci construisent des modèles probabilistes, des arbres de décision, des Séparateurs à Large Marge (SVM) ou encore des réseaux de neurones [9].

KEA (Witten *et al.*, 1999) est une méthode qui utilise une classification naïve bayésienne pour attribuer un score de vraisemblance à chaque terme candidat, le but étant d'indiquer s'ils sont des termes-clés ou non [10]. Witten *et al.* (1999) utilisent trois distributions conditionnelles apprises à partir du corpus d'apprentissage. La première correspond à la probabilité pour que chaque terme candidat soit étiqueté *oui* (terme-clé) ou *non* (non terme-clé). Les deux autres correspondent à deux différents traits qui sont le poids TF-IDF du terme candidat et sa première position dans le document :

$$P(\text{terme}) = \frac{P_{\text{oui}}(\text{terme})}{P_{\text{oui}}(\text{terme}) + P_{\text{non}}(\text{terme})} \tag{7}$$

$$P_{\text{oui}}(\text{terme}) = P(\text{terme}|\text{oui}) \times \prod_{\text{trait}\in\{\text{TF-IDF,position}\}} P_{\text{trait}}\,(\text{trait}(\text{terme})|\text{oui})$$

$$P_{\text{non}}(\text{terme}) = P(\text{terme}|\text{non}) \times \prod_{\text{trait}\in\{\text{TF-IDF,position}\}} P_{\text{trait}}\,(\text{trait}(\text{terme})|\text{non})$$

L'un des avantages de la classification naïve bayésienne est que chaque distribution est supposée indépendante. L'ajout de nouveaux traits dans la méthode KEA est donc très aisé.

Parmi les variantes de KEA proposées, Frank *et al.* (1999) ajoutent un troisième trait : le nombre de fois que le terme candidat est un terme-clé dans le corpus d'apprentissage. L'ajout de ce trait permet d'améliorer les performances de la version originale de KEA, mais uniquement lorsque la quantité de données d'apprentissage est très importante. Une autre amélioration de KEA, proposée par Turney (2003), tente d'augmenter la cohérence entre les termes candidats les mieux classés. Pour ce faire, une première étape de classification est effectuée avec la méthode originale. Cette première étape permet d'obtenir un premier classement des termes candidats selon leur score de vraisemblance. Ensuite, de nouveaux traits sont ajoutés et une nouvelle étape de classification est lancée. Les nouveaux traits ont pour but d'augmenter le score de vraisemblance des termes candidats ayant un fort lien sémantique avec certains des termes les mieux classés après la première étape. Enfin, Nguyen et Kan (2007) proposent l'ajout des

9. Sarkar *et al.* (2012) proposent une étude comparative de l'usage des arbres de décision, de la classification naïve bayésienne et des réseaux de neurones pour l'extraction automatique de termes-clés.

10. Il est important de noter que le score de vraisemblance pour chaque terme candidat permet aussi de les ordonner entre eux.

informations concernant la structure des documents. En effet, certaines sections telles que l'introduction et la conclusion dans les articles scientifiques sont plus susceptibles de contenir des termes-clés qu'une section présentant des résultats expérimentaux, par exemple. Dans leur version modifiée de KEA, ils proposent aussi l'usage de traits linguistiques tels que les parties du discours qui ont prouvées jouer un rôle non-négligeable pour l'extraction de termes-clés (Hulth, 2003).

En même temps que KEA (Witten *et al.*, 1999), Turney (1999) met au point l'algorithme génétique GenEx. GenEx est constitué de deux composants. Le premier composant, le géniteur, sert à apprendre des paramètres lors de la phase d'apprentissage. Ces paramètres sont utilisés par le second composant, l'extracteur, pour donner un score d'importance à chaque terme candidat. Plus les paramètres sont optimaux, meilleure est la classification des termes. Pour ce faire, les paramètres sont représentés sous la forme de bits qui constituent une population d'individus que le géniteur fait évoluer jusqu'à obtenir un état stable correspondant aux paramètres optimaux.

Dans son article présentant GenEx, Turney (1999) discute une autre méthode pour l'extraction de termes-clés. Cette méthode utilise de nombreux traits qui servent à entraîner 50 arbres de décision C4.5 (technique de *Random Forest*). Dans un arbre de décision, chaque branche représente un test sur l'un des traits d'un terme candidat. Les tests permettent un routage du terme candidat vers la feuille de l'arbre qui détermine sa classe. Grâce à la technique de *Random Forest*, soit l'usage de plusieurs arbres entraînés sur un échantillon différent du corpus d'apprentissage, l'extraction automatique de termes-clés est réduite à un vote de chaque arbre pour chaque terme candidat. Cela permet un classement des termes candidats en fonction de leur nombre de votes positifs. Les termes-clés extraits correspondent aux termes candidats les mieux classés.

La même année que les travaux de Hulth (2003) sur le bien fondé d'utiliser des traits linguistiques pour l'extraction automatique de termes-clés, Sujian *et al.* (2003) proposent une méthode utilisant un modèle d'entropie maximale (cf. équation 8) dont l'un des traits repose sur les parties du discours des mots qui composent les termes candidats. Un modèle de maximum d'entropie consiste à trouver parmi plusieurs distributions, une pour chaque trait, laquelle a la plus forte entropie. La distribution ayant la plus forte entropie est par définition celle qui contient le moins d'informations, ce qui la rend de ce fait moins arbitraire pour l'extraction des termes-clés.

$$\text{Score}(\text{terme}) = \frac{P(\text{oui}|\text{terme})}{P(\text{non}|\text{terme})} \tag{8}$$

$$P(\text{classe}|\text{terme}) = \frac{\exp\left(\sum_{\text{trait}} \alpha_{\text{trait}} \times \text{trait}(\text{terme}, \text{classe})\right)}{\sum_{c \in \{\text{oui},\text{non}\}} \exp\left(\sum_{\text{trait}} \alpha_{\text{trait}} \times \text{trait}(\text{terme}, c)\right)}$$

Le paramètre α_{trait} définit l'importance du trait auquel il est associé.

Les Séparateurs à Large Marge sont aussi des classifieurs utilisés par les méthodes d'extraction automatique de termes-clés. Ils exploitent divers traits afin de projeter des exemples et des contres-exemples sur un plan, puis ils cherchent l'hyperplan qui les sépare. Cet hyperplan sert ensuite dans l'analyse de nouvelles données. Dans le contexte de l'extraction de termes-clés, les exemples sont les termes-clés et les contres-exemples sont les termes candidats qui ne sont

pas des termes-clés. Ce mode de fonctionnement des SVM est utilisé par Zhang *et al.* (2006), mais un autre type de SVM est plus largement utilisé dans les méthodes supervisées d'extraction de termes-clés. Il s'agit de SVM qui utilisent de multiples marges représentant des rangs. Ces classifieurs permettent donc d'ordonnancer les termes-clés lors de leur extraction (Herbrich *et al.*, 1999; Joachims, 2006; Jiang *et al.*, 2009). La méthode KeyWE de Eichler et Neumann (2010) utilise ce type de SVM avec le trait TF-IDF ainsi qu'un trait booléen ayant la valeur vraie si le terme candidat apparaît dans un titre d'un article Wikipedia (un terme candidat apparaissant dans le titre d'un article de Wikipedia a une plus forte probabilité d'être un terme-clé). L'ordonnancement des termes candidats par le SVM permet ensuite de contrôler le nombre de termes-clés à extraire (choix des k termes candidats les mieux classés).

Tout comme Turney (1999), Ercan et Cicekli (2007) utilisent eux aussi une forêt d'arbres C4.5 dans leur méthode d'extraction de termes-clés. Ils utilisent des traits classiques et leur contribution se situe au niveau de l'utilisation d'un trait calculé à partir de chaînes lexicales. Une chaîne lexicale lie les mots d'un document selon certaines relations telles que la synonymie, l'hyponymie ou la méronymie. Ces relations permettent de calculer un score qui sert de trait. Cette approche est intéressante, mais du fait de limitations des chaînes lexicales actuellement disponibles elle présente l'inconvénient de ne retourner que des mots (aucun multi-mot). Cependant, l'usage d'une forêt d'arbre C4.5 permet un classement des mots à partir de leur nombre de votes positifs. Il est donc envisageable de déduire les termes-clés à partir de la liste ordonnée et pondérée des mots clés (voir les méthodes non-supervisées à bases de graphe – section 2.1).

Une autre méthode pour l'extraction automatique de termes-clés consiste à utiliser un perceptron multi-couches (Sarkar *et al.*, 2010). Un perceptron multi-couches est un réseau de neurones constitué d'au moins trois couches, chaque couche étant composée de neurones. Dans les deux couches extrêmes les neurones représentent respectivement les entrées et les sorties. Les couches centrales sont des couches cachées qui permettent d'acheminer les valeurs des entrées vers les sorties, où de nouvelles valeurs sont obtenues grâce à la pondération des transitions d'un neurone d'une couche vers un neurone de la couche suivante. Les entrées correspondent aux traits d'un terme candidat (ici TF-IDF, la position, la taille, etc.) et les sorties représentent les classes qu'il peut prendre (terme-clé ou non terme-clé). La valeur obtenue pour chaque sortie (classe) permet d'obtenir une probabilité pour que le terme candidat analysé soit un terme-clé ou non. Dans leur méthode, Sarkar *et al.* (2010) utilisent cette probabilité pour ordonner les termes candidats afin de mieux contrôler le nombre de termes-clés à extraire.

Dans leurs travaux, Liu *et al.* (2011) proposent une méthode d'extraction de termes-clés basée sur un modèle génératif. Leur méthode est très différente de celle de Witten *et al.* (1999) puisqu'ils décident d'utiliser une approche de traduction automatique. L'usage original de cette approche est justifié par le fait qu'un ensemble de termes-clés doit décrire de manière synthétique le document. Leur hypothèse est donc qu'un ensemble de termes-clés est une traduction d'un document dans un autre langage. Le modèle est appris à partir de paires de traductions dont l'un des termes est issu des titres ou des résumés des documents du corpus d'apprentissage et dont l'autre terme est issu des corps de ces mêmes documents. Les titres et les résumés sont utilisés comme langage synthétique et les corps des documents comme le langage naturel de ceux-ci.

3 Conclusion et perspectives

L'extraction automatique de termes-clés est une tâche importante qui permet la valorisation d'un document (représentation synthétique, mise en évidence des points clés dans le document, etc.) et qui facilite l'accès aux documents pertinents pour une requête utilisateur (indexation pour la recherche d'information).

Les méthodes existantes pour la tâche d'extraction automatique de termes-clés sont soit supervisées, soit non-supervisées. Les méthodes non-supervisées sont des méthodes émergentes ayant la particularité de s'abstraire de la spécificité des données traitées. Cette abstraction s'explique par des approches basées sur des constatations à propos de ce qu'est un terme-clé au sens général : importance sémantique, degré d'information, structure syntaxique, etc. Contrairement aux méthodes non-supervisées, les méthodes supervisées n'utilisent pas de propriétés définies à partir des traits statistiques et linguistiques, mais elles utilisent des modèles de décision appris à partir de ces traits, calculés sur les termes-clés d'un corpus d'apprentissage. L'usage d'un corpus d'apprentissage implique que les modèles appris soient spécifiques au domaine disciplinaire et à la langue de celui-ci. Cette spécificité peut s'avérer avantageuse lorsque le domaine et la langue que représente le corpus sont les mêmes pour les documents qui sont ensuite analysés, mais si tel n'est pas le cas les résultats de l'extraction peuvent en pâtir.

De futurs travaux peuvent se focaliser sur une hybridation des méthodes non-supervisées et supervisées. Dans un premier temps, il peut être intéressant de tenter d'améliorer les méthodes à base de graphe existantes. En effet, le graphe possède plusieurs points de variabilité sur lesquels il est possible d'agir pour affiner l'extraction : la granularité des nœuds, le type de relations permettant la création des arcs ou encore le facteur d'atténuation λ utilisé dans le calcul du score des nœuds. La granularité peut être étendue à des groupes de phrasèmes similaires (des variantes dont le sens est sensiblement le même). Cette nouvelle granularité peut impliquer la définition d'une nouvelle relation pour la création des arcs entre les nœuds. Enfin, des traits peuvent être appris, pondérés grâce à de l'apprentissage préalable, puis utilisés avec le facteur $(1 - \lambda)$ dans le calcul du score pour chaque nœud (voir la modification du score dans TopicalPageRank (Liu *et al.*, 2010)). Il est possible que ce dernier point demande de modifier la formule du score PageRank afin d'utiliser le score de recommandation et de nouveaux traits de manière cohérente (sans que la valeur d'un trait ne puisse annihiler le score de recommandation).

Remerciements

Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant la référence (ANR-12-CORD-0029).

Références

BARKER, K. et CORNACCHIA, N. (2000). Using Noun Phrase Heads to Extract Document Keyphrases. *In Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence : Advances in Artificial Intelligence*.

BOUDIN, F. et MORIN, E. (2013). Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression. *In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

BRIN, S. et PAGE, L. (1998). The Anatomy of a Large-Scale hypertextual Web Search Engine. *In Proceedings of the 7th International Conference on World Wide Web*.

CLAVEAU, V. (2012). Vectorisation, Okapi et Calcul de Similarité pour le TAL : pour Oublier Enfin le TF-IDF. *In Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2 : TALN*.

DING, Z., ZHANG, Q. et HUANG, X. (2011). Keyphrase Extraction from Online News Using Binary Integer Programming. *In Proceedings of 5th International Joint Conference on Natural Language Processing*.

EICHLER, K. et NEUMANN, G. (2010). DFKI KeyWE : Ranking Keyphrases Extracted from Scientific Articles. *In Proceedings of the 5th International Workshop on Semantic Evaluation*.

ERCAN, G. et CICEKLI, I. (2007). Using Lexical Chains for Keyword Extraction.

FRANK, E., PAYNTER, G., WITTEN, I., GUTWIN, C. et NEVILL-MANNING, C. (1999). Domain-Specific Keyphrase Extraction.

HASAN, K. et NG, V. (2010). Conundrums in Unsupervised Keyphrase Extraction : Making Sense of the State-of-the-Art. *In Proceedings of the 23rd International Conference on Computational Linguistics : Posters*.

HERBRICH, R., GRAEPEL, T. et OBERMAYER, K. (1999). Support Vector Learning for Ordinal Regression. *In Artificial Neural Networks, 1999*.

HULTH, A. (2003). Improved Automatic Keyword Extraction Given More Linguistic Knowledge. *In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

JIANG, X., HU, Y. et LI, H. (2009). A Ranking Approach to Keyphrase Extraction. *In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

JOACHIMS, T. (2006). Training Linear SVMs in Linear Time. *In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*.

JONES, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval.

JONES, S. et STAVELEY, M. (1999). Phrasier : a System for Interactive Document Retrieval Using Keyphrases. *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*.

KIM, S. N., MEDELYAN, O., KAN, M. et BALDWIN, T. (2010). Semeval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles. *In Proceedings of the 5th International Workshop on Semantic Evaluation*.

LIANG, W., HUANG, C., LI, M. et LU, B. (2009). Extracting Keyphrases from Chinese News Articles Using Textrank and Query Log Knowledge. *In Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*.

LITVAK, M. et LAST, M. (2008). Graph-Based Keyword Extraction for Single-Document Summarization. *In Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*.

LIU, Z., CHEN, X., ZHENG, Y. et SUN, M. (2011). Automatic Keyphrase Extraction by Bridging Vocabulary Gap. *In Proceedings of the 15th Conference on Computational Natural Language Learning*.

LIU, Z., HUANG, W., ZHENG, Y. et SUN, M. (2010). Automatic Keyphrase Extraction via Topic Decomposition. *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

LIU, Z., LI, P., ZHENG, Y. et SUN, M. (2009). Clustering to Find Exemplar Terms for Keyphrase Extraction. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1*.

MATSUO, Y. et ISHIZUKA, M. (2004). Keyword Extraction from a Single Document Using Word Co-occurrence Statistical Information.

MEDELYAN, O. et WITTEN, I. (2008). Domain-Independent Automatic Keyphrase Indexing with Small Training Sets.

MIHALCEA, R. et TARAU, P. (2004). Textrank : Bringing Order Into Texts. *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

NGUYEN, T. et KAN, M. (2007). Keyphrase Extraction in Scientific Publications. *In Proceedings of the 10th international conference on Asian digital libraries : looking back 10 years and forging new frontiers*.

NOBATA, C., COTTER, P., OKAZAKI, N., REA, B., SASAKI, Y., TSURUOKA, Y., TSUJII, J. et ANANIADOU, S. (2008). Kleio : a Knowledge-enriched Information Retrieval System for Biology. *In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*.

PAROUBEK, P., ZWEIGENBAUM, P., FOREST, D. et GROUIN, C. (2012). Indexation Libre et Contrôlée d'Articles Scientifiques Présentation et Résultats du Défi Fouille de Textes DEFT2012.

PAUKKERI, M. et HONKELA, T. (2010). Likey : Unsupervised Language-Independent Keyphrase Extraction. *In Proceedings of the 5th International Workshop on Semantic Evaluation*.

ROBERTSON, S. E., WALKER, S., BEAULIEU, M. et WILLETT, P. (1999). Okapi at TREC-7 : Automatic Ad Hoc, Filtering, VLC and Interactive Track.

SARKAR, K., NASIPURI, M. et GHOSE, S. (2010). A New Approach to Keyphrase Extraction Using Neural Networks.

SARKAR, K., NASIPURI, M. et GHOSE, S. (2012). Machine Learning Based Keyphrase Extraction : Comparing Decision Trees, Naïve Bayes, and Artificial Neural Networks.

SUJIAN, L., HOUFENG, W., SHIWEN, Y. et CHENGSHENG, X. (2003). News-Oriented Keyword Indexing with Maximum Entropy Principle.

TOMOKIYO, T. et HURST, M. (2003). A Language Model Approach to Keyphrase Extraction. *In Proceedings of the ACL 2003 workshop on Multiword expressions : analysis, acquisition and treatment-Volume 18*.

TURNEY, P. (1999). Learning Algorithms for Keyphrase Extraction.

TURNEY, P. (2003). Coherent Keyphrase Extraction via Web Mining.

WAN, X. et XIAO, J. (2008a). Collabrank : Towards a Collaborative Approach to Single-Document Keyphrase Extraction. *In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*.

WAN, X. et XIAO, J. (2008b). Single Document Keyphrase Extraction Using Neighborhood Knowledge. *In Proceedings of Association for the Advancement of Artificial Intelligence*.

WAN, X., YANG, J. et XIAO, J. (2007). Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction. *In Annual Meeting-association For Computational Linguistics*.

WITTEN, I., PAYNTER, G., FRANK, E., GUTWIN, C. et NEVILL-MANNING, C. (1999). KEA : Practical Automatic Keyphrase Extraction. *In Proceedings of the 4th ACM conference on Digital libraries*.

ZHANG, K., XU, H., TANG, J. et LI, J. (2006). Keyword Extraction Using Support Vector Machine.

Influence de l'étiquetage syntaxique des têtes sur l'analyse en dépendances discontinues du français

Ophélie Lacroix[1]
(1) LINA - Université de Nantes, 2 Rue de la Houssinière, 44322 Nantes Cedex 3
ophelie.lacroix@univ-nantes.fr

RÉSUMÉ

Dans cet article nous souhaitons mettre en évidence l'utilité d'un étiquetage syntaxique appliqué en amont d'une analyse syntaxique en dépendances. Les règles de la grammaire catégorielle de dépendances du français utilisées pour l'analyse gèrent les dépendances discontinues et les relations syntaxiques à longue distance. Une telle méthode d'analyse génère un nombre conséquent de structures de dépendances et emploie un temps d'analyse trop important. Nous voulons alors montrer qu'une méthode locale d'étiquetage peut diminuer l'ampleur de ces difficultés et par la suite aider à résoudre le problème global de désambiguïsation d'analyse en dépendances. Nous adaptons alors une méthode d'étiquetage aux catégories de la grammaire catégorielle de dépendance. Nous obtenons ainsi une pré-sélection des têtes des dépendances permettant de réduire l'ambiguïté de l'analyse et de voir que les résultats locaux d'une telle méthode permettent de trouver des relations distantes de dépendances.

ABSTRACT

On the Effect of Head Tagging on Parsing Discontinuous Dependencies in French

In this paper we want to show the strong impact of syntactic tagging on syntactic dependency parsing. The rules of categorial dependency grammar used to parse French deal with discontinuous dependencies and long distance syntactic relations. Such parsing method produces a substantial number of dependency structures and takes too much parsing time. We want to show that a local tagging method can reduce these problems and help to solve the global problem of dependency parsing disambiguation. Then we adapt a tagging method to types of the categorial dependency grammar. We obtain a dependency-head pre-selection allowing to reduce parsing ambiguity and to see that we can find distant relation of dependencies through local results of such method.

MOTS-CLÉS : Analyse syntaxique en dépendances discontinues, Étiquetage syntaxique.

KEYWORDS: Discontinuous Dependency Parsing, Syntactic Tagging.

1 Introduction

L'analyse syntaxique est une tâche bien connue dans le domaine du traitement automatique du langage naturel, permettant d'obtenir des structures syntaxiques à partir de phrases du langage naturel. On oppose couramment les représentations syntaxiques des structures par consituants et des structures en dépendances. Ici, nous nous intéressons particulièrement à la représentation en dépendances de ces structures (Tesnière, 1959; Mel'cuk, 1988). En utilisant cette représentation, nous souhaitons exprimer correctement les relations syntaxiques existantes entre les mots d'une phrase. Ces relations sont des relations binaires (dépendances) entre un gouverneur g et un subordonné s où le type de dépendance d est la fonction syntaxique existante entre g et s ($g \xrightarrow{d} s$). Une telle dépendance est projective si chaque mot dans l'intervalle [g,s] dépend de g (sinon elle est discontinue). Le type de dépendance d est aussi la dépendance-tête [1] du subordonné s. Notre travail se situe au niveau de l'analyse syntaxique en dépendances pour le français. Or cette langue admet des cas de discontinuité à travers des relations de longue distance comme la coréférence (voir figure 1) ou la comparaison ou des relations locales fréquentes, par exemple de négation ou de clitique. Nous avons choisi une méthode d'analyse guidée par les règles d'une

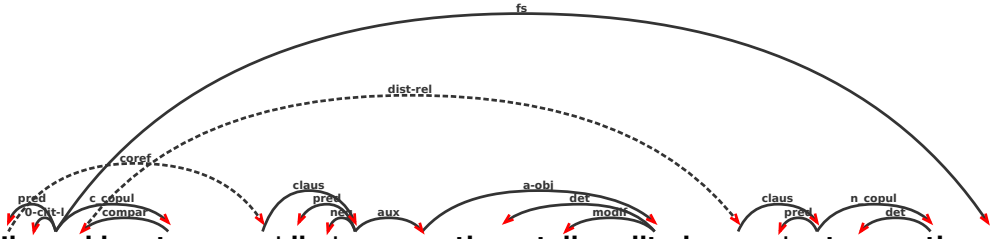


FIGURE 1 – *Structure de dépendances pour la phrase* "Il y a si longtemps qu'il n'a ressenti une telle solitude que c'est un vertige.". *Les dépendances projectives sont représentées par des lignes plaines tandis que les dépendances discontinues sont représentées par des lignes pointillées. Les types des dépendances sont les types utilisés par une grammaire catégorielle de dépendances du français.*

grammaire permettant d'obtenir des structures de dépendances projectives et des structures de dépendances discontinues [2]. Le modèle de grammaire catégorielle de dépendances (Dikovsky, 2004; Béchet *et al.*, 2005; Dekhtyar et Dikovsky, 2008; Dekhtyar *et al.*, 2012) étend la gestion des dépendances aux dépendances discontinues et est donc tout à fait adaptée à la représentation syntaxique en dépendances de phrases du français. Le CDG Lab (Alfared *et al.*, 2011) est un outil, destiné à l'analyse syntaxique avec des grammaires catégorielles de dépendances et au développement de corpus arborés en dépendances. Il propose trois modes d'analyses différents que nous redéfinirons par la suite. Le mode nous intéressant ici est le mode semi-automatique de *sélection des têtes*. Dans ce mode, un utilisateur souhaitant procéder à une analyse syntaxique en

1. La dépendance-tête est le type de la dépendance arrivant sur le subordonné.

2. Une structure de dépendances discontinue est une structure dans laquelle on trouve au moins une dépendance discontinue. Dans ces structures les dépendances peuvent se croiser. Par exemple, les clitiques engendrent des dépendances discontinues dès lors qu'une forme composée verbale est employée, séparant le verbe et son objet cliticisé. La négation produit fréquemment une discontinuité puisqu'elle est communément composé de deux particules, parfois distantes ("Ne ... que"), parfois inversés ("Jamais ... ne"). Par ailleurs, la relation de coréférence (figure 1) est intraphrasale, elle correspond à la co-prédication définie par (Mel'cuk, 1988).

dépendances pourra sélectionner manuellement les dépendances-têtes. Cette sélection des têtes en amont de l'analyse syntaxique améliore grandement la vitesse d'analyse par rapport à une analyse automatique à partir des phrases brutes du français. Nous souhaitons donc remplacer, pour notre travail, cette sélection des têtes manuelles par une sélection automatique. Cette tâche est similaire à celle d'étiquetage grammaticale ou d'étiquetage morphosyntaxique. L'idée d'utiliser un pré-étiquetage pour réduire l'ambiguïté et améliorer une analyse syntaxique en dépendances à déjà été exploitée dans ce cadre (Nasr, 2006; Candito *et al.*, 2010). Ici, nous souhaitons mettre en place un procédé de type *supertagging* (Bangalore et Joshi, 2010b). La sélection des têtes est en fait un étiquetage syntaxique des unités lexicales des phrases du français adapté à la grammaire catégorielle de dépendance du français utilisée pour l'analyse en dépendance. Il ne s'agit donc pas d'apporter des informations grammaticales ou morphosyntaxiques (propres aux unités lexicales) à l'analyseur mais bien d'apporter des informations syntaxiques qui définissent des fonctions binaires entre unités lexicales. La difficulté est alors de trouver les bonnes étiquettes syntaxiques de manière locale, avec des informations locales, bien que la fonction syntaxique auquelle l'étiquette réfère concerne deux unités lexicales potentiellement distantes. Nous procéderons alors dans un premier temps à cet étiquetage syntaxique en utilisant la méthode des CRF [3] adaptée aux types de dépendances de la grammaire catégorielle de dépendances du français. Nous essayons ici d'utiliser une méthode locale pour résoudre un problème global. Il s'agit de la principale difficulté de cette méthode. Nous souhaitons donc voir si elle permettra d'obtenir les bonnes dépendances-têtes. Puis nous exécuterons l'analyse en dépendances sur les phrases ainsi étiquetées pour constater l'effet positif de cet étiquetage sur le temps d'analyse et sur la production (les structures de dépendances sortantes) de l'analyseur. Pour conclure, nous nous questionnerons sur la place de cette méthode dans une analyse totalement autonome. Est-elle suffisante vis-à-vis des résultats obtenus ou peut-elle être associée à d'autres procédés permettant de combler les imperfections de celle-ci ?

Ce travail s'inscrit dans un travail de plus grande envergure qui comprendra un travail de découpage des phrases en unités lexicales, ainsi que leur étiquetage grammaticale, un travail d'étiquetage syntaxique précédant celui d'analyse syntaxique en dépendances, puis finira par un travail concernant le tri des structures de dépendances en sortie de l'analyseur. Ici nous nous intéressons en particulier à l'étiquetage syntaxique. Nous supposons donc avoir en entrée un bon découpage des phrases en unités lexicales composées ainsi qu'un bon étiquetage grammatical de ces unités.

2 Grammaires catégorielles de dépendances

Le modèle de grammaires catégorielles de dépendances est un modèle de grammaires similaire aux grammaires catégorielles classiques (Bar-Hillel *et al.*, 1964) auxquelles est ajoutée la notion de valence polarisée permettant d'introduire les dépendances discontinues. Dans ce modèle, les catégories sont des types de dépendances et permettent de représenter les dépendances projectives tandis que les valences polarisées sont des types de dépendances associées à des polarités duales ($\nearrow\searrow$et $\swarrow\nwarrow$) permettant de représenter les dépendances discontinues (voir (Dekhtyar et Dikovsky, 2008)). Les règles utilisées par cette classe de grammaires sont présentées dans la figure 1. Les structures de dépendances produites à l'aide de ces grammaires sont alors des graphes orientés acycliques.

3. *Conditional Random Fields* en anglais ou champs markovien conditionnels en français.

$\mathbf{L}^1$	$C^{P_1}\left[C\backslash\beta\right]^{P_2} \vdash \left[\beta\right]^{P_1P_2}$
$\mathbf{I}^1$	$C^{P_1}\left[C^*\backslash\beta\right]^{P_2} \vdash \left[C^*\backslash\beta\right]^{P_1P_2}$
Ω^1	$\left[C^*\backslash\beta\right]^{P} \vdash \left[\beta\right]^{P}$
$\mathbf{D}^1$	$\alpha^{P_1(\swarrow C)P(\nwarrow C)P_2} \vdash \alpha^{P_1PP_2}$, si $(\swarrow C)(\nwarrow C)$ satisfait le principe **FA**

TABLE 1 – *Règles gauches des grammaires catégorielles de dépendances. Des règles symétriques sont utilisées dans le cas des dérivations à droite. Les règles **L, I** et Ω permettent d'éliminer les catégories classiques et les catégories itérables (i.e. dérivables infiniement), et de concaténer ou conserver les valences polarisées en une chaîne que l'on appelle* potentiel*. L'élimination des valences dans la dérivation (règle **D**) se fait sur le principe **FA** (First Available) : les valences duales les plus proches dans un potentiel sont éliminées en premier.*

2.1 Données de la grammaire catégorielle de dépendances du français

Pour notre travail, nous utiliserons une grammaire catégorielle de dépendances du français (Dikovsky, 2011). Elle est constituée d'un ensemble conséquent de règles elles-mêmes composées de types de dépendances (les catégories de la grammaire). Ces types de dépendances, 117 au total, représentent un vaste champ de fonctions syntaxiques exprimant les particularités du français. Ces nombreux types de dépendances sont rassemblés en 39 groupes de dépendances selon leurs fonctions syntaxiques. Par exemple, les dépendances de type objet accusatif (*a-obj*), objet datif (*d-obj*), objet génitif (*g-obj*) sont réunies dans le groupe des objets : *OBJ*. Par ailleurs, les types de dépendances peuvent être associés à des dépendances discontinues, on en compte 27. Parmi les types associés aux dépendances discontinues on trouve ceux appartenant aux groupes des clitiques (*CLIT*), des modifieurs (*MODIF*), des réflexifs (*REFLEX*), des coréférences (*COREF*) et appositions (*APPOS*), des éléments de négation (*NEG*), des aggrégations (*AGRR*), etc.

En outre, les règles de la grammaire sont associées à des classes grammaticales. Lors d'une analyse, avec le choix des règles se fait le choix de ces classes grammaticales et des traits morphologiques des unités lexicales établies en fonction des valeurs des traits employés par le Lefff[4] (Sagot, 2010). On dénombre 185 classes grammaticales.

2.2 CDG Lab : Analyseur en dépendances

Le CDG Lab (Alfared *et al.*, 2011) est un outil de travail dédié à l'analyse en dépendances guidée par les règles de grammaires catégorielles de dépendances. L'analyseur en dépendances du CDG Lab propose 3 modes d'analyse différents mais complémentaires :

– *l'analyse autonome* est un mode permettant de lancer l'analyse à partir d'une phrase du français sans indiquer manuellement d'informations complémentaires. La phrase est donc découpée en mots qui sont eux-mêmes réassociés en unités lexicales possibles[5]. L'analyse (basée sur un algorithme CYK modifié) est alors exécutée à partir de ce découpage.
– *l'analyse par sélection des têtes* est un mode semi-automatique. Avant de procéder à l'analyse, l'utilisateur à la possibilité de choisir les bonnes unités lexicales, leurs classes grammaticales et leurs dépendances-têtes. L'analyse peut ensuite être lancée en tenant compte de ces choix.

4. Lexique des formes fléchies du français.
5. Basées sur Lefff (Sagot, 2010).

– *l'analyse par approximation* s'effectue à la suite d'une analyse automatique ou d'une analyse par sélection des têtes. Elle permet d'annoter positivement ou négativement les attributions des classes grammaticales et/ou des types de dépendances. Appliqué autant de fois que nécessaire, ce mode permet de raffiner la production de l'analyse : la(les) structure(s) de dépendances résultante(s).

Le mode qui nous intéresse ici est celui de la sélection des têtes en amont de l'analyse syntaxique en dépendances. Nous savons que choisir manuellement les dépendances-têtes d'une phrase avant analyse permet de réduire l'ambiguïté en faisant converger l'analyse vers un ensemble de solutions plus restreint. Les avantages se remarquent au niveau du temps de calcul de l'analyse et au niveau de la production de l'analyseur, celui-ci produisant moins de structures de dépendances en sortie. Notons que la sélection des têtes peut se faire au niveau des types de dépendances ou des groupes de dépendances. On appellera alors respectivement : **dépendance-tête** ou **groupe-tête**, le type ou le groupe de dépendances sélectionné pour une unité lexicale. Nous souhaitons donc remplacer la sélection manuelle des têtes par une sélection automatique et comprendre l'apport réel de cette tâche avec un algorithme standard d'apprentissage.

2.3 Corpus en dépendances, données grammaticales et syntaxiques

Le corpus que nous utiliserons pour nos expérimentations a été annoté en dépendances semi-automatiquement grâce à l'outil CDG Lab. Il est composé de 2778 structures de dépendances associées à des phrases du français provenant de registres variés et comprenant au total 35203 unités lexicales composées. Les dépendances discontinues représentent 4% du nombre total de dépendances du corpus et elles sont présentes (au moins une fois) dans 41% des structures de dépendances.

Les types de dépendances utilisés dans la représentation de ces structures correspondent aux types de la grammaire catégorielle de dépendances du français. De plus, chaque unité lexicale dans le corpus est annotée correctement par une classe grammaticale. Pour procéder à l'étiquetage syntaxique les données utilisées seront :

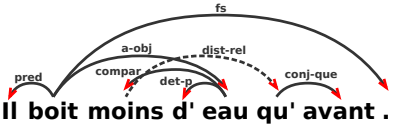
– les unités lexicales composées
– les dépendances-têtes (ou groupes-têtes)
– les classes grammaticales

Le nombre de classes grammaticales étant important, nous décidons de sous-catégoriser ces classes pour arriver à deux formes de sous-classification : les classes grammaticales simples (28 classes) et les classes grammaticales étendues (86 classes). Une classe grammaticale simple indique la classe grammaticale d'une unité lexicale sans autre information tandis qu'une classe grammaticale étendue ajoute des informations (parfois sémantiques) potentiellement utiles syntaxiquement. Les classes grammaticales étendues, plus précises, permettent de mieux cibler les types et groupes de dépendances comme exposé dans la table 2.

Nombre moyen de		types	(max.)	groupes	(max.)
Par classe	simple	13	(43)	7	(18)
grammaticale	étendue	6	(31)	4	(16)

TABLE 2 – *Nombre moyen (et maximum) de types et groupes de dépendances possibles par classe grammaticale simple ou étendue.*

Un exemple, regroupant les données par groupe/type de dépendances et par classe grammaticale simple et étendue, est donné par la figure 2.



Unité lexicale	Dépendance-tête type	groupe	Classe grammaticale simple	étendue
Il	pred	PRED	PN	PN-pers-n
boit	S	SENT	Vt	Vt-fin
moins	compar	COMPAR	Adv	Adv-degr-compar
d'	det-p	DET	Det	Det
eau	a-obj	OBJ	N	N
qu'	dist-rel	REL	Conj	Conj-comp
avant	conj-que	CONJ	Adv	Adv
.	fs	PUNCT	FullStop	FullStop

FIGURE 2 – *Structure de dépendances et tableau rapportant les dépendances-têtes, les groupes-têtes, les classes grammaticales simples et les classes grammaticales étendues de chaque unité lexicale de la phrase* "Il boit moins d'eau qu'avant." *Les classes grammaticales étendues ajoutent des informations en plus de la classe. Par exemple,* Adv-degr-compar *et* Conj-comp *indique un adverbe et une conjonction qui sont tous deux impliqués dans une comparaison. Pour ce travail, nous n'avons pas utilisé les traits de Lefff.*

3 Étiquetage syntaxique

Le problème de l'étiquetage est un problème largement étudié dans le domaine du traitement automatique de la langue naturelle. Les tâches d'étiquetage grammatical ou morphosyntaxique sont les plus répandues mais diffèrent de l'étiquetage syntaxique. Néanmoins les outils restent les mêmes. Parmi les méthodes existantes pour accomplir la tâche d'étiquetage syntaxique on trouvera, les modèles graphiques probabilistes tels que les modèles de Markov cachés (HMM) (Rabiner, 1989), les modèles d'entropie maximale (MEMM) (Ratnaparkhi, 1996) et les champs markoviens conditionnels (CRF) (Sutton et McCallum, 2006; Lafferty *et al.*, 2001). Pour notre travail, nous avons choisi d'utiliser ces derniers car ils permettent de prendre en compte plus d'informations que les HMM et qu'ils sont bien adaptés à l'attribution de séquences d'étiquettes alors que les MEMM sont plus performants pour la classification.

3.1 Logiciel et patrons de traits

Nous avons choisi le logiciel Wapiti (Lavergne *et al.*, 2010) pour entraîner un modèle et étiqueter syntaxiquement notre corpus car il est capable de travailler avec un grand nombre d'étiquettes . Il utilise les CRF pour cet entraînement et attribue donc des poids à des traits choisis. Ces traits peuvent être extraits à partir de patrons de traits définis à l'avance. Le logiciel nous laisse la

possibilité de lui fournir des patrons de traits modifiables que nous avons testés.
Comme indiqué dans la partie 2.3, chaque phrase du corpus est décomposée en unités lexicales elles-mêmes étiquetées grammaticalement (par des classes grammaticales simples ou étendues selon le choix d'expérimentation). Nous disposons donc de ces informations. Nous pouvons choisir une largeur de fenêtre (appliquée autour d'une unité lexicale) pour indiquer si l'on tient compte des unités lexicales et des classes grammaticales précédentes et suivantes lors de l'assignation d'une étiquette syntaxique. Nous constatons qu'une fenêtre de 5 (2 mots avant, 2 mots après) donne de bons résultats, qu'élargir la fenêtre à 7 pour les unités lexicales génère beaucoup de traits pour peu d'améliorations mais qu'élargir la fenêtre à 7 autour des classes grammaticales est beaucoup plus efficace. Il est aussi intéressant d'associer unité lexicale et classe grammaticale dans un même trait. Les premiers patrons de traits choisis sont les suivants :

Unité lexicale courante
Unité lexicale précedente de 1
Unité lexicale précedente de 2
Unité lexicale suivante de 1
Unité lexicale suivante de 2
Classe grammaticale de l'unité lexicale courante
Classe grammaticale de l'unité lexicale précedente de 1
Classe grammaticale de l'unité lexicale précedente de 2
Classe grammaticale de l'unité lexicale précedente de 3
Classe grammaticale de l'unité lexicale suivante de 1
Classe grammaticale de l'unité lexicale suivante de 2
Classe grammaticale de l'unité lexicale suivante de 3
Unité lexicale courante et sa classe grammaticale

Nous testons aussi quelques traits comme l'extraction du suffixe des unités lexicales (testé pour 2, 3 ou 4 lettres) et le fait de savoir si une unité lexicale commence par une majuscule et retenons les suivants :

Suffixe de 3 lettres de l'unité lexicale courante
L'unité lexicale précédente commence-t-elle par une majuscule ?

Notons ici que les traits choisis sont toujours des traits unigrammes, les traits bigrammes générant trop de traits non pertinants. Cependant, pour chaque trait unigramme, la probabilité qu'il apparaîsse avec chacune des étiquettes possibles est calculée lors de l'apprentissage. L'ensemble de ces patrons de traits génère alors plus d'un million [6] de traits pour chaque modèle d'apprentissage et permet d'obtenir de bons résultats d'étiquetage précisés dans la section suivante.

3.2 Expérimentations et Évaluation

Pour procéder à l'étiquetage syntaxique nous avons divisé le corpus (voir section 2.3) en 10 parties égales. Chaque partie est étiquetée selon un modèle entraîné sur les 9 autres parties. L'entraînement se fait sur des données parfaitement étiquetées grammaticalement et syntaxiquement. La possibilité de choisir des données plus ou moins informatives (classe grammaticale simple ou étendue ; dépendance-tête ou groupe-tête) permet de réaliser 4 expérimentations

6. L'ensemble des patrons de traits génère, au pire, plus de 32000 traits unigrammes différents qui associés aux différentes possibilités d'étiquettes (au maximum 117 pour les types) produit jusqu'à 3,7 millions de traits.

différentes. De plus, l'outil Wapiti nous permet d'engendrer les n meilleurs étiquetages pour une séquence donnée. Nous avons donc choisi de produire les 10 meilleurs étiquetages syntaxiques pour chaque phrase d'entrée. Ainsi à chaque expérimentation, nous récupérons 10 séquences d'étiquettes pour chaque phrase du corpus. Ces séquences sont potentiellement assez similaires. Souvent, seulement quelques étiquettes varient d'une séquence à une autre. Pour évaluer la qualité de l'étiquetage syntaxique nous considérons les 1, 2, 5 ou 10 meilleures étiquettes de chaque unité lexicale de chaque phrase du corpus. Les résultats de l'évaluation sont présentés dans la table 3.

Étiquetage des dépendances-têtes

	Classes Grammaticales simples				Classes Grammaticales étendues			
	Pré.	(Moy.)	Rap.	(Moy.)	Pré.	(Moy.)	Rap.	(Moy.)
Top 1	87.8	(70.7)	87.8	(62.9)	91.1	(77.8)	91.1	(70.1)
Top 2	83.4	(66.0)	90.0	(67.5)	86.5	(72.5)	93.2	(74.1)
Top 5	73.0	(56.2)	92.9	(73.4)	75.1	(61.3)	95.5	(79.6)
Top 10	62.9	(46.6)	94.6	(77.2)	63.6	(51.0)	96.6	(82.4)

Étiquetage des groupes-têtes

	Classes Grammaticales simples				Classes Grammaticales étendues			
	Pré.	(Moy.)	Rap.	(Moy.)	Pré.	(Moy.)	Rap.	(Moy.)
Top 1	90.4	(86.5)	90.4	(80.1)	91.6	(89.5)	91.6	(85.6)
Top 2	85.6	(81.0)	92.5	(83.6)	86.8	(83.8)	93.7	(87.9)
Top 5	74.3	(67.7)	95.1	(87.9)	75.0	(71.8)	96.0	(91.2)
Top 10	63.3	(55.2)	96.4	(90.6)	63.4	(57.8)	97.1	(93.1)

TABLE 3 – *Évaluation de l'étiquetage syntaxique produit par Wapiti. D'une part, la précision et le rappel sont calculés globalement sur toutes les étiquettes. La précision est le nombre d'étiquettes correctes sur le nombre d'étiquettes différentes attribuées. Le nombre d'étiquettes différentes attribuées varie selon le top, il peut y en avoir 1, de 1 à 2, de 1 à 5 ou de 1 à 10 (on ne compte pas deux fois la même étiquette). Le rappel est le nombre d'unités lexicales pour lesquelles on a trouvé la bonne étiquette (parmi les 1, 2, 5 ou 10 étiquettes attribuées) sur le nombre d'étiquettes du corpus d'entrée (i.e. le nombre d'unités lexicales). D'autre part, une moyenne de la précision et du rappel sur les types/groupes de dépendances est aussi calculée (entre parenthèses). Dans ce cas, pour chaque type/groupe, la précision est le nombre d'étiquettes correctement attribuées sur le nombre d'étiquettes différentes attribuées pour ce type/groupe. Le rappel pour un type/groupe est le nombre d'unités lexicales y appartenant pour lesquelles on a trouvé la bonne étiquette sur le nombre d'étiquettes de ce type/groupe existantes dans le corpus d'entrée.*

Un permier constat face aux résultats d'étiquetage est de voir l'utilité des informations apportées par les classes grammaticales étendues. De ce côté les résultats sont meilleurs en précision et en rappel. De manière plus approfondie, on peut voir que plus on considère d'étiquettes plus la précision diminue tandis que le rappel augmente. En effet plus on a d'étiquettes différentes pour une unité lexicale plus on a de chance d'avoir la bonne étiquette parmi celles-ci mais on ne sait pas de laquelle il s'agit, on perd donc en précision. En fait, les résultats par étiquette varient grandement. Parmi les 2, 5 ou 10 séquences d'étiquettes pour une même phrase, seulement quelques étiquettes varient. Les étiquettes qui ne changent pas (ou peu) à chaque séquence sont globalement "sûres" et perdent peu en précision (comme les déterminants *DET*, la ponctuation

PUNCT, la négation *NEG* dans le cas des groupes). Celles qui gagnent fortement en rappel sont celles qui sont souvent mal attribuées dans la première séquence mais que l'on finit par trouver dans les suivantes (comme les relations souvent distantes de coréférence *COREF* ou d'apposition *APPOS*). Nous souhaitons voir quel impact a ce gain en rappel sur l'analyse syntaxique en dépendance. Dans la section suivante nous verrons dans quelle mesure l'étiquetage syntaxique réduit le temps d'analyse en dépendance et permet d'obtenir une meilleure structure de dépendances selon les différents critères d'étiquetage que nous avons établis auparavant.

4 Analyse syntaxique en dépendances et évaluation

4.1 Procédure d'analyse et d'évaluation

Pour procéder à l'analyse syntaxique en dépendances nous souhaitons adapter l'outil d'analyse par sélection des têtes du CDG Lab pour assigner automatiquement les étiquettes syntaxiques, trouvées par Wapiti, en tant que dépendances-têtes (ou groupes-têtes). Nous attribuons donc 1, 1 à 2, 1 à 5 ou 1 à 10 types (ou groupes) de dépendances différents à chaque unité lexicale composée selon les résultats des top 1, 2, 5 et 10 de l'étiquetage syntaxique. Nous utilisons ici les meilleurs résultats, c'est à dire ceux trouvés avec les classes grammaticales étendues. L'analyse syntaxique en dépendances guidée par les règles de la grammaire catégorielle de dépendances du français s'éxecute en tenant compte des différentes dépendances-têtes (ou groupes-têtes) possibles. Le CDG Lab est conçu pour produire une liste des structures de dépendances possibles pour chaque analyse [7]. La sélection des têtes permet de réduire l'ambiguïté en contraignant l'analyseur à chercher des structures de dépendances dont les types sont en accord avec cette sélection. Vis-à-vis d'une analyse autonome, ici, le nombre de structures de dépendances en sortie est moindre. Nous observons donc des temps d'analyse également réduits. Les structures de dépendances en sortie de l'analyseur ne sont pas triées. Or nous souhaitons avant tout savoir si parmi les structures de dépendances produites pour une phrase donnée se trouve la bonne structure de dépendances (i.e. la structure de dépendances associée à cette phrase dans le corpus en dépendances de référence, 2.3). L'idée est donc de trier ces structures de la plus proche à la plus éloignée de la structure originale. La plus proche étant celle ayant le plus de dépendances en commun [8] avec la structure de référence. Les différentes étapes de ce traitement sont illustrées dans la figure 3.

Nous nous intéressons alors seulement à la première structure de dépendances de chaque liste (la plus proche de la structure originale). Néanmoins, parfois, il n'existe aucune structure de dépendances produite. Deux raisons sont possibles :

– les dépendances-têtes (ou groupes-têtes) assignées sont en contradiction avec les règles de la grammaire, cela entraîne alors un échec de l'analyse ;
– le temps d'analyse est trop élevé (communément due à la longueur de la phrase), l'analyse s'interrompt donc avant d'aboutir.

Nous souhaitons donc connaître le nombre de structures de dépendances obtenues en sortie. Les résultats de ses expérimentations sont présentés dans la section suivante.

7. En accord avec les dépendances-têtes (ou groupes-têtes) sélectionnées ainsi qu'avec la grammaire.

8. Une dépendance est commune aux deux structures de dépendances si elle possède dans les deux structures le même gouverneur, le même subordonné et le même type de dépendance.

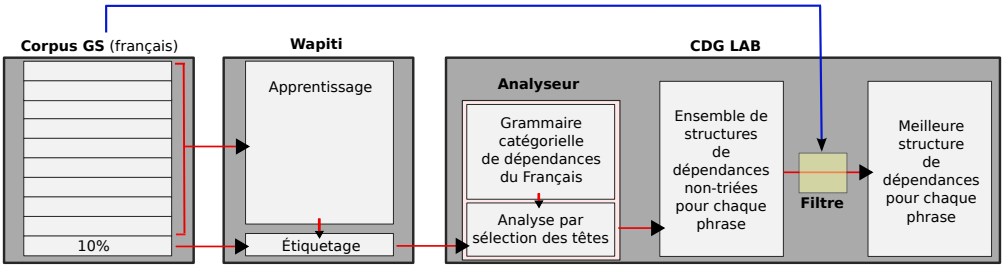


FIGURE 3 – *Schéma expliquatif du traitement complet. Wapiti est utilisé pour procéder à l'apprentissage sur 90% du corpus et à l'étiquetage sur 10%. La partie étiquetée est analysée par l'analyseur du CDG Lab en tenant compte des dépendances-têtes (ou groupes-têtes). On obtient plusieurs structures de dépendances pour chaque phrase qui sont ensuite triées selon leur conformité avec la structure originale du corpus de référence (filtre). Le traitement est opéré sur chaque partie du corpus.*

4.2 Résultats et discussions

Les premiers résultats exposés dans la table 4 présente les taux d'analyses abouties ainsi que le nombre d'unités lexicales par phrase et le temps de calcul.

Nous rapportions dans la section 3.2 qu'en ayant plus de choix de têtes pour chaque unité lexicale nous avions plus de chance d'obtenir la bonne tête parmi ceux-ci. Il en est de même pour les dépendances lorsqu'on laisse entre 1 à 10 choix de têtes pour chaque unité lexicale : ayant plus de chance d'avoir les bonnes dépendances-têtes (ou groupes-têtes) nous avons aussi plus de chance d'obtenir une structure de dépendances proche de la structure de dépendances de référence parmi toutes celles produites. Pour la même raison, on obtient de meilleurs taux d'analyses ayant abouti. Dans le meilleur des cas (sélection de 10 étiquettes), nous avons 2548 analyses sur 2778 (91.7%) ayant abouti dont 2088 ayant trouvé, parmi les structures de dépendances produites, une structure de dépendances entièrement correcte. Les temps d'analyse augmentent relativement à l'ambiguïté (en obtenant plus de structures de dépendances en sortie) ainsi qu'à la longueur des phrases. Effectivement les phrases qui sont analysées avec un choix de dix étiquettes alors qu'elles ne l'étaient pas avec un choix inférieur sont souvent plus longues car plus difficiles à étiqueter correctement et exploitent plus de temps d'analyse. Par ailleurs, on constate que le nombre moyen d'unités lexicales par phrase augmente légèrement dans le cas des analyses ayant abouti quand le choix d'étiquettes est plus large. Ce qui montre que des phrases plutôt longues qui n'ont pas été analysées avec un seul choix d'étiquettes l'ont été avec plus de choix. Cependant un nombre d'étiquettes plus important en entrée augmente l'ambiguïté de l'analyse et donc le temps d'analyse. On obtient donc un peu plus de phrases non-analysées par manque de temps lorsqu'on augmente le choix d'étiquettes.
Lorsque l'on compare les résultats des expérimentations faites avec les dépendances-têtes et les groupes-têtes, on constate plusieurs points intéressants. Les taux d'analyses abouties sont meilleurs lorsqu'on utilise les groupes-têtes car on laisse un plus large choix à l'analyseur (les groupes comprennent parfois plusieurs types de dépendances). Néanmoins on note une différence au niveau du temps d'analyse qui est inférieur lorsqu'on utilise les dépendances-têtes. En effet, l'analyseur converge plus vite. On peut donc voir qu'il y a moins d'analyses n'ayant pas abouti par manque de temps mais que le nombre d'analyses n'ayant pas abouti car étant non-conforme

Analyse autonome

Nb de têtes	Nombre de phrases AA	(%)	NA-C	(%)	NA-T	(%)	UL/phrase AA	NA-C	NA-T	Temps d'analyse AA	NA
0	1150	(41.4)	3	(00.1)	1625	(58.5)	7.2	7.3	17.6	42min24	4h30

Analyse avec sélection des dépendances-têtes

Nb de têtes	Nombre de phrases AA	(%)	NA-C	(%)	NA-T	(%)	UL/phrase AA	NA-C	NA-T	Temps d'analyse AA	NA
1	1805	(65.0)	969	(34.9)	4	(00.1)	11.5	16.5	52.5	3min03	1min35
1 à 2	2054	(73.9)	718	(25.8)	6	(00.2)	11.6	17.7	56.1	4min16	1min53
1 à 5	2335	(84.1)	438	(15.8)	5	(00.2)	12.0	20.0	49.4	6min02	1min29
1 à 10	2505	(90.2)	262	(09.4)	11	(00.4)	12.2	22.5	42.8	8min01	2min23

Analyse avec sélection des groupes-têtes

Nb de têtes	Nombre de phrases AA	(%)	NA-C	(%)	NA-T	(%)	UL/phrase AA	NA-C	NA-T	Temps d'analyse AA	NA
1	1931	(69.5)	832	(29.9)	15	(00.5)	11.5	16.9	45.8	6min41	3min31
1 à 2	2172	(78.2)	586	(21.1)	20	(00.7)	11.6	18.6	45.0	8min52	4min15
1 à 5	2439	(87.8)	302	(10.9)	37	(01.3)	11.8	21.6	43.6	12min05	6min47
1 à 10	2548	(91.7)	179	(06.4)	51	(01.8)	12.0	24.4	41.6	16min43	9min03

TABLE 4 – *Calcul du nombre de phrases dont l'analyse a abouti (AA), du nombre de phrases dont l'analyse n'a pas abouti car elle est non-conforme à la grammaire (NA-C), du nombre de phrases dont l'analyse n'a pas abouti par manque de temps (NA-T). Le temps d'analyse est limité à 10s maximum. Calcul du nombre moyen d'unités lexicales (UL) par phrase dont l'analyse a abouti et dont l'analyse n'a pas abouti (car non-conforme ou par manque de temps). Calcul du temps total d'analyse pour celles ayant abouti et celles n'ayant pas abouti.*

à la grammaire est plus élevé. En fait, l'étiquetage est plus précis donc plus rapide mais conduit plus facilement à une analyse non-conforme à la grammaire s'il y a une ou plusieurs étiquettes fausses. On obtient plus facilement une incohérence vis-à-vis de la grammaire.

D'autre part, nous présentons dans la table 5 en tant que score de précision, les scores d'attachement obtenus sur les analyses abouties. On y trouve le pourcentage d'unités lexicales pour lesquelles le bon gouverneur et la bonne étiquette ont été trouvés (LAS) et le taux d'unités lexicales pour lesquelles le bon gouverneur a été trouvé (UAS). Encore une fois, nous pouvons voir que plus large est le choix d'étiquettes en entrée plus les scores sont meilleurs. Ils atteignent globalement des taux élevés et sont quelques peu meilleurs dans le cas où l'on considère seulement l'exactitude du gouverneur. Lorsqu'on s'intéresse aux dépendances discontinues, on remarque que la précision sur ces dépendances est légèrement moins bonne que sur l'ensemble des dépendances.

La différence de précision entre les analyses ayant reçu des dépendances-têtes ou des groupes-têtes est négligeable sur l'ensemble des dépendances mais moins bonne sur les dépendances discontinues dans le cas de la sélection des groupes-têtes. Cela peut s'expliquer par le fait que le taux de dépendances discontinues est moins élevé parmi les dépendances des analyses abouties dans le cas de la sélection des dépendances-têtes [9]. Les cas difficiles de dépendances discontinues distantes sont écartés du calcul de la précision si la pré-sélection des têtes est non-conforme à la

9. On obtient de 4,3% à 4,6% de dépendances discontinues parmis les dépendances des analyses abouties avec sélection des dépendances-têtes pour 4,8% à 4,9% avec la sélection des groupes-têtes.

grammaire et engendre une mauvaise analyse. Nous avons vu que cette non-conformité est plus facile à atteindre avec la sélection des dépendances-têtes. Les scores sont donc moins bons avec la sélection des groupes-têtes car plus d'analyse aboutissent sans forcement avoir résolu les cas discontinus difficiles.

Analyse autonome

Nb de têtes	Toutes dépendances		Dépendances discontinues	
	LAS	UAS	LAS	UAS
0	98.3	99.0	92.7	93.2

Analyse avec sélection des dépendances-têtes

Nb de têtes	Toutes dépendances		Dépendances discontinues	
	LAS	UAS	LAS	UAS
1	93.7	96.7	92.4	93.7
1 à 2	95.1	97.3	94.3	95.5
1 à 5	96.2	97.8	94.4	95.5
1 à 10	96.4	97.9	94.5	95.4

Analyse avec sélection des groupes-têtes

Nb de têtes	Toutes dépendances		Dépendances discontinues	
	LAS	UAS	LAS	UAS
1	93.9	96.7	88.8	93.3
1 à 2	95.1	97.2	90.0	93.7
1 à 5	96.3	97.9	90.5	93.8
1 à 10	96.7	98.0	91.1	94.3

TABLE 5 – *Évaluation de l'analyse autonome (sans pré-sélection des têtes) et de l'analyse avec pré-sélection des dépendances-têtes et des groupes-têtes. Cette évaluation est réalisée sur la meilleure structure de dépendances produite par l'analyseur (i.e. la plus proche de la structure de dépendances de référence) pour chaque analyse aboutie. Les scores d'attachement LAS et UAS correspondent respectivement au score d'attachement avec dépendances étiquetées* (Labeled Attachment Score) *et au score d'attachement avec dépendances non-étiquetées* (Unlabeled Attachment Score). *Ils sont calculés sur toutes les dépendances d'une part et sur les seules dépendances discontinues d'autre part, en excluant les dépendances liées à des signes de ponctuations dans les deux cas.*

4.3 Travaux reliés

Plusieurs travaux ont déjà mis en évidence l'utilité des méthodes de type *supertagging* sur l'analyse syntaxique (Clark et Curran, 2004; Sarkar, 2010). Les résultats de ces travaux sont difficiles à comparer avec d'autres pour plusieurs raisons. D'une part les fonctions syntaxiques utilisées ici pour l'analyse en dépendances diffèrent et sont plus nombreuses que dans les travaux où les dépendances proviennent des têtes de constituants. De plus les dépendances discontinues ne sont pas toujours prises en compte. D'autres part, l'analyse en dépendances n'est ici pas totalement autonome en s'appuyant sur certains pré-requis. Nous pouvons tout de même tenter de rapprocher ces travaux d'autres tâches de *supertagging* pour l'anglais (Nasr et Rambow, 2004) ou pour l'allemand (Foth *et al.*, 2006). Les travaux les plus proches sont sans doute ceux de (Nasr et Rambow, 2010) obtenant une précision de 85,7% pour l'anglais.

5 Conclusion et travaux à venir

Les résultats de l'analyse syntaxique en dépendances contrainte par la sélection des têtes reflète d'une réelle utilité de cette sélection automatique. Dans un premier temps, la sélection des dépendances-têtes ou des groupes-têtes en amont de l'analyse en dépendances permet de réduire de manière significative le temps d'analyse. De nombreuses phrases, d'une longueur conséquente, ne permettant pas d'aboutir à une analyse autonome peuvent être finalement analysées grâce à la sélection des têtes. Ce facteur est très important pour atteindre des taux de réussite (analyses abouties) intéressant et donc des résultats réellement exploitables. Par ailleurs, en étiquetant syntaxiquement les unités lexicales des phrases de 1 à 10 étiquettes différentes, on obtient un bon score en précision. Il nous indique que parmi les structures de dépendances produites par l'analyseur du CDG Lab, on obtient très souvent la bonne structure de dépendances pour une phrase donnée.

Cependant, nous supposons ici avoir un bon découpage des phrases en unités lexicales et un bon étiquetage en entrée ainsi qu'un tri des structures de dépendances en sortie qui s'appuie sur la structure de dépendances de référence. Dans l'idée de mettre en place un analyseur totalement autonome, nous souhaitons, par la suite, faire de ces étapes des tâches automatiques. Nous avons donc l'intention d'ajouter une étape de découpage des unités lexicales et d'étiquetage grammatical de ces unités en amont de l'étiquetage syntaxique présenté dans cet article. Puis nous appliquerons une méthode de tri automatique des structures de dépendances en sortie de l'analyseur permettant de trouver la structure de dépendances la plus proche de la structure de référence sans s'y être référé.

En outre, notons que le taux d'analyse non abouties car l'étiquetage était non-conforme avec la grammaire varie de 6 à 35% du meilleur au pire des cas. Un mauvais étiquetage local peut être la cause de cette non-conformité. Cependant le score général d'étiquetage étant bon (le meilleur est de 97.1 en rappel pour 10 choix d'étiquettes), il est évident que la majorité des étiquettes pour une phrase donnée sont correctes et permettraient d'obtenir une (ou plusieurs) sous-structure(s) de dépendances correcte(s) pour cette phrase. L'évolution de l'analyseur du CDG Lab ira dans ce sens : permettre à l'analyseur de produire des structures de dépendances partielles lorsque la sélection des têtes n'est pas totalement conforme avec la grammaire. La solution partielle pourra ensuite être complétée en appliquant une analyse par approximation. Le nombre de structures de dépendances analysées augmentera et cela permettra d'obtenir de meilleurs taux d'analyses abouties.

Références

ALFARED, R., BÉCHET, D. et DIKOVSKY, A. (2011). "CDG Lab" : a Toolbox for Dependency Grammars and Dependency Treebanks Development. *In Proceedings of DEPLING 2011*, pages 272–281.

BANGALORE, S. et JOSHI, A., éditeurs (2010a). *Complexity of Lexical Descriptions and its Relevance to Natural Language Processing : A Supertagging Approach*. MIT Press.

BANGALORE, S. et JOSHI, A. K. (2010b). *Supertagging : Using Complex Lexical Descriptions in Natural Language Processing*. Mit Press.

BAR-HILLEL, Y., GAIFMAN, C. et SHAMIR, E. (1964). On Categorial and Phrase Structure Grammars. *In Language and information*, pages 99–115. Addison-Wesley.

BÉCHET, D., DIKOVSKY, A. et FORET, A. (2005). Dependency structure grammar. *In Proceedings of LACL 2005*, pages 18–34.

CANDITO, M., CRABBÉ, B. et DENIS, P. (2010). Statistical french dependency parsing : treebank conversion and first results. *In Proceedings of LREC 2010*, pages 1840–1847.

CLARK, S. et CURRAN, J. R. (2004). The importance of supertagging for wide-coverage ccg parsing. *In Proceedings of COLING 2004*, pages 282–288.

DEKHTYAR, M. et DIKOVSKY, A. (2004). Categorial dependency grammars. *In Proceedings of Intern. Conf. on Categorial Grammars*, pages 76–91.

DEKHTYAR, M. et DIKOVSKY, A. (2008). Generalized categorial dependency grammars. *In Trakhtenbrot/Festschrift*, LNCS 4800, pages 230–255. Springer.

DEKHTYAR, M., DIKOVSKY, A. et KARLOV, B. (2012). Iterated dependencies and kleene iteration. *In Formal Grammar 2010/2011*, LNCS 7395, pages 66–81.

DIKOVSKY, A. (2004). Dependencies as categories. *In Proceedings of COLING 2004 Workshop, "Recent Advances in Dependency Grammars"*, pages 90–97.

DIKOVSKY, A. (2011). Categorial dependency grammars : from theory to large scale grammars. *In DEPLING 2011*.

FOTH, K., BY, T. et MENZEL, W. (2006). Guiding a constraint dependency parser with supertags. *In Proceedings of COLING 2006*, pages 289–296.

LAFFERTY, J., MCCALLUM, A. et PEREIRA, F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *In Proceedings of ICML 2001*.

LAVERGNE, T., CAPPÉ, O. et YVON, F. (2010). Practical very large scale CRFs. *In ACL 2010*.

MEL'CUK, I. (1988). *Dependency syntax : Theory and Practice*. State University of New York Press.

NASR, A. (2006). Grammaires de dépendances génératives probabilistes. modèle théorique et application à un corpus arboré du français. *Traitement Automatique des Langues*, 46(1):115–153.

NASR, A. et RAMBOW, O. (2004). Supertagging and full parsing. *In Proceedings of TAG+7*.

NASR, A. et RAMBOW, O. (2010). Non-lexical chart parsing for tag. *In* (Bangalore et Joshi, 2010a).

RABINER, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *In Proceedings of IEEE 1989*.

RATNAPARKHI, A. (1996). A maximum entropy model for part-of-speech tagging. *In Proceedings of EMNLP 1996*.

SAGOT, B. (2010). The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. *In Proceedings of LREC 2010*.

SARKAR, A. (2010). Combining supertagging and lexicalized tree-adjoining grammar parsing. *In* (Bangalore et Joshi, 2010a).

SUTTON, C. et MCCALLUM, A. (2006). An introduction to conditional random fields for relational learning. *In Introduction to Statistical Relational Learning*. MIT Press.

TESNIÈRE, L. (1959). *Éléments de syntaxe structurale*. Klincksieck.

Une approche linguistique pour l'extraction des connaissances dans un texte arabe

Houda Saadane
LIDILEM, Université Stendhal – Grenoble III, 1180, avenue central, F-38400 Saint Martin d'Hères
houda.saadane@e.u-grenoble3.fr

RÉSUMÉ

Nous présentons dans cet article un système d'extraction de connaissances en arabe, fondé sur une analyse morphosyntaxique profonde. Ce système reconnaît les mots simples, les expressions idiomatiques, les mots composés et les entités nommées. L'analyse identifie aussi les relations syntaxiques de dépendance et traite les formes passives et actives. L'extraction des connaissances est propre à l'application et utilise des règles d'extraction sémantiques qui s'appuient sur le résultat de l'analyse morphosyntaxique. A ce niveau, le type de certaines entités nommées peut être révisé. L'extraction se base, dans nos expérimentations, sur une ontologie dans le domaine de la sécurité. Le RDF (Resource Description Framework) produit est ensuite traité pour regrouper les informations qui concernent un même événement ou une même entité nommée. Les informations ainsi extraites peuvent alors aider à appréhender les informations contenues dans un ensemble de textes, alimenter une base de connaissances, ou bien servir à des outils de veille.

ABSTRACT

A linguistic approach for knowledge extraction from an Arabic text

We present in this paper a knowledge extraction system for Arabic. The information extraction is based on a deep morphosyntactic analysis. It also recognizes single words, idiomatic expressions, compounds and named entities. The analysis also identifies dependency relations, verb tenses and passive/active forms. Information extraction is application-independent and uses extraction rules that rely on the result of the morphosyntactic analysis. At this level, some named entity categories can be reconsidered. This extraction is based in our experimentations on the security ontology. The Resource Description Framework (RDF) obtained is then processed to gather information concerning a single event or named entity. The information extracted can help to understand the information contained in a set of texts, to infer knowledge into a knowledge base, or be used for monitoring tools.

MOTS-CLÉS : Analyse linguistique, fouille de textes, arabe, entités nommées, extraction d'informations, règles d'extraction, ontologie.

KEYWORDS : Linguistic analysis, Text Mining, Arabic, named entities, information extraction, extraction rules, ontology.

1 Introduction

Les évolutions rapides des nouvelles technologies sont accompagnées d'un essor important

de la quantité d'information disponible sur le Web, et nécessitent le développement d'outils pour analyser et structurer les documents textuels. Ainsi, les documents en arabe sur le Web, à l'instar des auteurs langues, se multiplient en nombre, en contenus et en quantité. Pour traiter cette grande masse de textes, une possibilité est de recourir à des outils de fouille de textes ou d'extraction de connaissances. Pourtant, dans ce domaine en pleine émergence, les outils qui permettent d'analyser les documents arabes se limitent en général à l'extraction d'entités nommées.

L'objectif de cet article est de présenter un système d'extraction des connaissances pour l'arabe qui, après avoir effectué une analyse linguistique profonde du texte, extrait les entités nommées, et les relie à des événements, en se basant sur une ontologie métier. Le développement d'un tel système exige l'étude des propriétés des données textuelles en soulevant essentiellement les problèmes concernant l'analyse et la représentation des contenus des textes (Cherfi., 2002).

Dans cet article, nous commençons par présenter dans la section 2 un état de l'art du domaine de l'extraction de connaissances, suivi dans la section 3 d'une description des étapes de l'analyse linguistique profonde. La section 4 décrit le processus d'extraction sémantique qui se base sur les résultats de l'analyse syntaxique et sur des concepts et des règles d'extraction spécifiques au domaine traité. Dans la section 5, nous présentons le traitement qui consiste à aligner les informations recueillies au niveau du document pour regrouper les occurrences qui désignent une même entité nommée, ou un même événement. Nous exposons dans la section 6 les premiers résultats que nous avons obtenus. Nous terminons par une ouverture sur les perspectives d'amélioration de notre système.

2 Etat de l'art

De nombreux travaux de recherche ont abordé la question de l'extraction de connaissances, citons TEMIS (TExt MIning Solutions) qui est un éditeur de logiciels d'enrichissement sémantique des contenus. Sa plateforme Luxid (Kuznik et al., 2010) propose des fonctionnalités d'extraction d'information qui sont réalisées par le moteur Insight Discoverer Extractor. Ce serveur d'extraction d'information pour l'analyse des textes enchaine trois étapes : l'analyse de corpus (identification de la langue), l'analyse linguistique réalisée par Xelda et l'extraction des connaissances (identification des entités nommées, reconnaissance des relations entre les entités). Cette étape repose sur la technologie Skill Cartridge (Brun et al., 2009) qui propose un ensemble de règles et de composants linguistiques définissant l'information à extraire. La cartouche va permettre de réaliser une analyse sémantique pour fournir les relations sémantiques. Ces relations sémantiques font référence par la suite à des entités nommées et à des patrons spécifiques au domaine de l'intelligence économique. Un rôle est assigné à chaque entité, afin de définir sa position dans la relation sémantique. Les langues analysées par IDE, sont le français, l'anglais, l'espagnol, l'italien et l'allemand.

Le projet SAMAR[1] (Station d'Analyse Multimédia en langue Arabe) est, quant à lui, destiné aux journalistes travaillant en langue arabe. Son objectif est le développement d'une plateforme de traitement multimédia à destination de la presse et des médias arabes. Parmi les principales composantes de ce projet, il y a l'analyse sémantique et l'extraction des

[1] http://www.samar.fr/

entités nommées, qui constituent les sujets des informations à analyser. Ces entités sont par la suite stockées dans une base de connaissances.

Concernant plus particulièrement l'extraction des connaissances en langue arabe, les travaux se sont focalisés sur la reconnaissance et l'extraction des entités nommées. Parmi ces travaux, nous pouvons mentionner les travaux de (Zitouni et al., 2005) qui utilisent des techniques d'apprentissage automatique (Modèles de Markov à Entropie Maximale) en considérant des jeux de descripteurs idoines. L'utilisation d'un corpus parallèle pour l'extraction des entités nommées en arabe a été adoptée par (Samy et al., 2005). Cette méthode est aussi basée sur des règles, mais avec en plus l'utilisation d'un lexique monolingue de langue espagnole pour permettre l'extraction des entités nommées en espagnol. Une fois que ces entités sont extraites, un processus de transcription en arabe est appliqué sur ces entités.

Enfin, le travail de thèse de Mesfar (2008) avait pour objectif une analyse morpho-syntaxique et une extraction des entités nommées en arabe standard. Son système est basé sur une combinaison des résultats obtenus par le biais d'un analyseur morphologique et de grammaires locales représentant des règles d'identification écrites à la main. D'autres travaux récents se basent sur des méthodes à base de règles Shaalan et al., 2009 ; Zaghouani et al., 2010.

Les études que nous avons décrites ci-dessus proposent une solution à la problématique de l'extraction des entités nommées, mais n'abordent pas, ou peu, le sujet de l'extraction de connaissances relatives à ces entités. Notre objectif est de proposer un système d'extraction de connaissances pour l'arabe, qui sera capable de repérer les entités nommées, mais aussi les relations sémantiques qui les relient, pour un domaine particulier modélisé dans une ontologie. Notre analyse est fondée sur la technologie des automates d'états finis.

3 Analyse linguistique profonde

L'analyse linguistique profonde est nécessaire pour assurer une extraction d'informations sûre, pertinente et complète, par exemple en reliant des éléments qui peuvent être éloignés dans la phrase initiale.
L'analyse que nous avons mise au point se divise en plusieurs étapes allant du découpage en mots jusqu'aux relations que ceux-ci entretiennent au sein d'une phrase. Les principales étapes de cette analyse sont décrites dans les sous-sections suivantes :

3.1 Découpage en mots

La tokenisation permet le découpage du texte en mots, les « tokens », séparés par des ponctuations ou par des espaces. Elle prend aussi en compte les balises, les dates abrégées, etc. Citons l'exemple de la tokenisation en mots de la phrase باريس مدينة الجن والملائكة . » (Paris la ville des diables et des anges) donnera : باريس | مدينة | الجن | و | الملائكة. C'est une étape qui va permettre d'attribuer ensuite à chaque token des catégories et des propriétés sur lesquelles portera l'analyse profonde.

3.2 Analyse morphologique

Le travail de l'analyseur morphologique consiste à retrouver la forme de surface d'un mot

stocké dans le lexique à partir de la forme canonique de ce dernier (infinitif du verbe, masculin singulier d'un adjectif, etc...). Cette étape est primordiale lors de l'analyse linguistique. Elle se divise à son tour en plusieurs étapes : la consultation du dictionnaire des formes fléchies d'une part pour récupérer la normalisation du mot et d'autre part, pour permettre de récupérer les informations linguistiques (genre, nombre, catégorie grammaticale, etc.) de ce mot. L'une des particularités de la langue arabe est la présence des formes agglutinées (formes avec des proclitiques et des enclitiques). Ces formes ne sont pas présentes dans le dictionnaire des formes fléchies. Pour identifier ces formes et les traiter correctement, nous avons ajouté un segmenteur de clitiques (proclitiques et enclitiques) à l'analyse morphologique. Cette segmentation des formes agglutinées se déroule de la manière suivante (Semmar et al., 2005) :

1. Recherche de toutes les compositions possibles entre les clitiques (proclitique, enclitique) et le radical en utilisant les dictionnaires des proclitiques, enclitiques et formes fléchies.

2. Chaque radical est ensuite recherché dans le dictionnaire des formes fléchies. Si ce radical n'existe pas dans le dictionnaire, des transformations morphologiques sont appliquées avant leur suffixation en se basant sur des règles de réécriture, enfin le radical résultat est de nouveau recherché dans le dictionnaire des formes fléchies. Par exemple, considérons la forme agglutinée «بسيارته» (avec sa voiture) et les clitiques inclus dans cette forme (ب, ه). Le radical récupéré «سيارت» n'existe pas dans le dictionnaire des formes fléchies. Mais après l'application de la règle de réécriture transformant la lettre «ت» en «ة» en fin de mot, le radical modifié «سيارة» (voiture) est trouvé dans le dictionnaire des formes fléchies et la forme agglutinée «بسيارته» est découpée en proclitique + radical + enclitique comme suit : بسيارته = ب + سيارة + ه (avec sa voiture).

3. Une étape supplémentaire permet de vérifier la relation d'ordre au sein d'une représentation des formants du mot sur un vecteur ordonné (Zmantar et al., 2009). La principale propriété de celui-ci est que chaque proclitique est incompatible avec un proclitique de même position, en raison de la relation d'ordre strict qui régit les formants du mot graphique. Exemples : wa et fa coordonnants (واو العطف et فاء), qui occupent tous les deux la même position sur le vecteur d'ordre, sont incompatibles entre eux (ils ne peuvent pas apparaître dans un même mot). Cette étape doit aussi vérifier les règles, syntaxiques mais aussi sémantiques, de compatibilité et d'incompatibilité entre les proclitiques et les enclitiques.

Cette analyse reconnaît aussi des expressions idiomatiques afin de grouper certains mots pour les considérer comme une seule unité (سكة الحديد : Chemin de fer). Cette reconnaissance se fait à l'aide de règles et de dictionnaires.

Si, après ces étapes, un mot reste inconnu, le système lui attribue une (des) catégorie(s) par défaut, en s'appuyant sur des informations révélées par sa forme de surface. Par exemple, s'il s'agit d'un mot en caractères latins majuscules, comme ONU, il sera étiqueté comme un nom propre.

Après cette analyse morphologique, et particulièrement pour le traitement de la langue arabe, la majorité des mots restent ambigus à cause de l'absence des voyelles courtes arabes

dans les textes (Debili et al., 1998), ce qui est moins prononcé pour les autres langues. Le problème majeur rencontré dans cette phase est celui de l'ambiguïté lexicale et grammaticale, qui découle du fait que lorsqu'un mot est reconnu, l'analyseur morphologique peut fournir plusieurs interprétations qui renvoient à plusieurs catégories syntaxiques ou à plusieurs sens. Le rôle du désambiguïseur morpho-syntaxique qui intervient par la suite, est de réduire le nombre des ambiguïtés grammaticales en utilisant des matrices de désambiguïsation. Ce sont des matrices de bi-grammes et tri-grammes de catégories obtenues à partir d'un corpus étiqueté du LDC (Arabic Treebank), et désambiguïsé manuellement. Le résultat de l'application des n-grammes nous permet d'obtenir la suite de couples mot-catégories la plus probable. L'ambiguïté lexicale est conservée à ce niveau, pour être traitée plus tard, au niveau de l'extraction sémantique.

3.3 Repérage des dates

Lors de l'analyse morphologique, un traitement spécifique intervient pour le repérage des dates. Ceci permet ensuite à la désambiguïsation d'être plus efficace, étant donné que les dates ne sont plus constituées d'une suite de catégories, mais sont associées à une catégorie « date ».

Les dates et heures se composent de l'indication normalisée du temps qu'elles représentent. Nous nous sommes basés sur la norme ISO 8601 avec le format AAAAMMJJ où AAAA représente l'année sur 4 chiffres, MM représente le numéro du mois, sur 2 chiffres et JJ représente le quantième dans le mois, sur 2 chiffres. Par exemple, «أفريل 1984 (Avril) 01» est normalisé de la manière suivante : «19840401 », «أفريل (Avril) 1984 » est normalisé par «198404XX», «غداً : demain» est normalisé par «XXXXXX+1».

3.4 Mots composés

Une étude linguistique spécifique de la langue arabe nous a permis de définir et d'écrire un certain nombre de règles dans le but d'établir des relations de dépendance (contiguës et non contiguës) entre les mots au sein du syntagme nominal. Ces relations permettent ensuite de reconnaître les mots composés présents dans une phrase.

Citons l'exemple de «أرملة الشهيد» (la veuve de martyr), nous avons une relation entre deux mots associés par annexion (معرف بالإضافة), qui relie le mot indéfini)veuve(أرملة et le mot défini الشهيد (martyr) pour donner une relation de type "NomRelNom".

3.5 Relations sujet-verbe-complément

Nous avons défini un certain nombre de règles, issues d'une étude expérimentale pour l'identification et le repérage des relations syntaxiques dans une phrase. Notons que certains verbes demandent un complément, contrairement à d'autres. Ces verbes sont appelés des verbes transitifs. Il faut définir la liste des verbes transitifs et des verbes intransitifs, étant donné que, en arabe, la position des mots ne suffit pas à en déduire la fonction syntaxique du mot. Les voyelles courtes l'indiquent, mais elles ne sont généralement pas indiquées dans les textes écrits. C'est pour cela qu'il faut se baser sur la transitivité ou sur la non transitivité du verbe pour déterminer quelles sont les relations qui existent entre un nom et un verbe.

Voici les relations que nous détectons :

- Les relations agent-verbe, qui permettent d'identifier l'agent de l'action (pour répondre à la question : qui a fait l'action?)
- Les relations verbe-complément, qui permettent d'identifier qui a subi l'action, ou encore les circonstanciels qui nous renseignent sur le moyen (comment? Avec quoi?), la date (quand?), le lieu (où ?), ... de l'action.

3.6 Passif

Afin de rendre l'étape de construction des règles d'extraction des connaissances plus efficace, l'analyse linguistique profonde adopte en interne la même représentation pour une phrase passive, et pour son équivalent à la forme active. Cette phase consiste donc à identifier les formes passives et à les transformer en formes actives. Voici quelques structures syntaxiques exprimant un passif (Ziad, 2010) :

- Passif avec un verbe doublement transitif. مُنِح الشاعرُ جائزةً : On a accordé un prix à l'écrivain
- Passif avec un verbe transitif indirect, précédé par une préposition, حُكِمَ عليه بالاعدام : Il a été condamné à mort.
- Emploi de tournures modernes du passif, qui expriment le complément d'agent : مِنْ قِبَل ، مِنْ طَرَفِ ، مِنْ جَانِب ، عَلَى يَد (par).

Dans l'exemple suivant : إعتقلت الفتاة على يد الشرطة (la fille a été arrêtée par la police), pour ne pas confondre entre la personne qui fait l'action et la personne qui la subit, il est important de savoir si la forme est active ou passive. Ici, la forme est active mais emploie une tournure moderne du passif qui exprime le complément d'agent (عَلَى يَد), donc le sujet est la police et le complément est la fille. Nous obtenons donc les relations suivantes :

- relation agent-verbe entre إعْتَقَلَ (arrêter) et شُرْطَةٌ (Police) reliés par le mot عَلَى يَد (par)
- relation verbe-complément entre إعْتَقَلَ (arrêter) et فَتَاة (fille).

```
<relation reltype="SV">
 <head>
  <posBeg>112</posBeg>
  <lemma>إعْتَقَلَ </lemma>
  <catPos index="no">+verbe</catPos>
  <prop
index="no">+vbpassif+acc+3fs</prop>
  <posEnd>118</posEnd>
 </head>
 <dept>
  <posBeg>133</posBeg>
  <lemma>شُرْطَةٌ</lemma>
  <catPos index="no">+nom</catPos>
  <prop index="no">+fs</prop>
  <posEnd>139</posEnd>
 </dept>
 <lingIndication index="no">
  <posBeg>126</posBeg>
```

```
<relation reltype="VC">
 <head>
  <posBeg>112</posBeg>
  <lemma>إعْتَقَلَ </lemma>
  <catPos index="no">+verbe</catPos>
  <prop
index="no">+vbpassif+acc+3fs</prop>
  <posEnd>118</posEnd>
 </head>
 <dept>
  <posBeg>119</posBeg>
  <lemma>فَتَاة</lemma>
  <catPos
index="no">+annppers</catPos>
  <prop index="no">+pers+fs</prop>
  <posEnd>125</posEnd>
 </dept>
</relation>
```

```
  <lemma>عَلَى يَد</lemma>
  <catPos index="no">+prepN</catPos>
  <prop index="no">+passif</prop>
  <posEnd>132</posEnd>
 </lingIndication>
</relation>
```

TABLE 1 – Exemple d'extraction des relations syntaxiques dans une phrase passive.

Head : unité qui constitue la tête de la relation
Dept : unité qui constitue le dépendant de la relation
LingIndication : balise qui contient des indications sur les unités qui permettent de relier les termes d'une relation, et qui serviront lors de l'extraction sémantique.

3.7 Reconnaissance des entités nommées

Cette phase consiste à mettre en œuvre un système de reconnaissance et de typage des entités nommées. Dans notre approche, nous avons opté pour un système à base de règles linguistiques qui exploitent l'étiquetage syntaxique, des marqueurs lexicaux (déclencheurs) et des dictionnaires de noms propres. La mise en place de règles de reconnaissance d'entités nommées a nécessité une recherche profonde sur certains traits linguistiques propres aux entités nommées en arabe.

Exemple : " الأخ مُعز غرسلاوي " (le frère Moez Garsallaoui) Dans cet exemple, nous avons le titre de civilité "الأخ : frère" suivi d'un prénom et d'un nom propre. Voici la représentation que nous obtenons :

```
<en entype="pers">
 <relation reltype="AnnpNP">
  <head>
   <posBeg>1104</posBeg>
   <lemma>غرسلاوي </lemma>
   <catPos index="no">+np</catPos>
   <posEnd>1111</posEnd>
  </head>
  <dept>
   <posBeg>1092</posBeg>
   <lemma>أَخٌ</lemma>
   <catPos ndex="no">+annppers</catPos>
   <prop index="no">+pers+ms</prop>
   <posEnd>1097</posEnd>
  </dept>
 </relation>
```

```
<relation reltype="PrenomNP">
  <head>
   <posBeg>1104</posBeg>
   <lemma>غرسلاوي </lemma>
   <catPos index="no">+np</catPos>
   <posEnd>1111</posEnd>
  </head>
  <dept>
   <posBeg>1098</posBeg>
   <lemma>مُعِز</lemma>
   <catPos
index="no">+prenom</catPos>
   <prop index="no">+m</prop>
   <posEnd>1103</posEnd>
  </dept>
 </relation>
</en>
```

TABLE 2 – Exemple de reconnaissance des entités nommées de type Personne.

4 Extraction sémantique

L'entrée de cette étape est constituée de la sortie de l'analyse morpho-syntaxique décrite précédemment. Cette analyse fournit les informations suivantes :

- les lemmes des mots ainsi que leur position dans le texte, et leur catégorie grammaticale
- les relations de dépendance syntaxique entre les mots,
- les entités nommées typées.

Nous avons choisi une représentation interlingue en anglais de toutes les informations, dans le but de faciliter la lecture des informations extraites par les non arabophones et de faciliter la fusion d'informations provenant de documents en plusieurs langues. Nous avons eu recours à deux types d'opérations : l'utilisation de dictionnaires de traduction existants et l'ajout d'un système de translittération pour les entités nommées qui n'existent pas dans les dictionnaires (Saadane et al., 2012).

L'extraction de connaissances permet de mettre en évidence des entités nommées et des relations relatives à un concept particulier, par exemple : « arrestation », « attentat », « condamnation », « construction ». Le déroulement de cette étape s'effectue en trois temps : la sélection des concepts potentiellement présents, la sélection des règles à appliquer, puis l'application des règles.

4.1 Sélection des concepts probables

Une étape primordiale lors de l'extraction des connaissances consiste à repérer les déclencheurs. Ces déclencheurs peuvent être des mots, des expressions ou des relations, et indiquent qu'une relation relative à un concept peut être présente dans le texte. Les déclencheurs sont présentés sous forme de deux colonnes :

- la première colonne contient les mots, les expressions ou encore les relations qui indiquent la présence du concept dans le texte traité.
- la deuxième colonne définit le concept (arrestation, transfert, émission, union,...) associé à l'élément de la première colonne.

إعْتَقَلَ Arrest
VC#نَقَلَ#رِسَالَة# Emission

Comme nous l'avons mentionné et comme le montre l'exemple précédent, les déclencheurs peuvent être des mots (إعتقل, زواج) :(mariage, interpeller...), des expressions ou bien des relations (VC# نَقَلَ # رِسَالَة #) : (VC#transmettre#message#).

Il est nécessaire qu'un déclencheur soit présent dans l'entrée afin d'être en mesure d'extraire l'information présente. A partir d'un déclencheur, un concept est obtenu ce qui nous permet ensuite de sélectionner les règles à appliquer.

4.2 Sélection des règles à appliquer

Les déclencheurs nous ont permis d'obtenir la liste des concepts présents dans le texte. Ces concepts nous amènent alors vers une liste de règles qui vont être confrontées aux relations issues de l'analyse syntaxique. Si les relations syntaxiques correspondent aux règles définies, elles pourront alors être extraites. La définition des règles à appliquer comporte aussi deux

colonnes :

- La première colonne indique les concepts. Ils sont ensuite comparés aux concepts probables sélectionnés à la phase précédente par le biais des déclencheurs
- La deuxième colonne liste les règles spécifiques à un concept pouvant être appliquées. Ces règles contiennent des relations syntaxiques (SV, VC) entre un verbe et une liste de catégories, avec d'éventuelles prépositions.

Arrest SV#إعْتَقَل#<en># عَلَى يَد (Arrest SV#arrêter#<en># par)
Arrest SV#إعْتَقَل#<en># (Arrest SV#arrêter#<en>#)
Arrest VC#إعْتَقَل|#<en># (Arrest VC#arrêter#<en>#)

Pour illustrer notre propos, prenons l'exemple suivant : إعتقلت العروض على يد الشرطة (Elaroud a été arrêtée par la police). Lors de la première étape, le mot «اعتقل» (arrêter) a été repéré comme déclencheur du concept « Arrest» (arrestation). Ce concept est associé aux règles présentées ci-dessus. Afin de pouvoir être sélectionnées, les règles doivent correspondre à une relation syntaxique présente dans la phrase. Or, dans la phrase « Elaroud a été arrêtée par la police », « arrêter » a un complément « Elaroud » de type entité nommée, et « arrêter » a un sujet « police », étant donné que la phrase est au passif. Donc, ce sont les règles SV#إعْتَقَل #<en>#عَلَى يَد et VC#إعْتَقَل #<en># qui seront sélectionnées.
A ce stade intervient un traitement qui permet l'application des règles sélectionnées lors de la phase précédente aux relations syntaxiques effectivement présentes, afin d'extraire l'information en question.

4.2.1 L'application des règles sélectionnées

Les règles sélectionnées à l'étape précédente sont appliquées aux relations syntaxiques afin d'en extraire les connaissances présentes. Les connaissances sont extraites à partir de la sortie de l'analyse linguistique profonde. La règle d'extraction indique ensuite quels sont les éléments qui doivent être extraits, et quelle est leur sémantique. Si nous reprenons l'exemple « إعتقلت العروض على يد الشرطة: Elaroud a été arrêtée par la police», l'une des règles sélectionnée est la relation verbe-complément entre «arrêter» et une entité nommée. Cette règle indique que le concept extrait sera «Arrest» (Arrestation), dont le patient est l'objet de «arrêter», c'est-à-dire «Elaroud». Le résultat obtenu sera alors de la forme suivante : *<gs:Arrest rdf:nodeID="id17Arrest"><wn:undergoer rdf:nodeID="id1Elaroud"/></gs:Arrest>*

4.2.2 L'extraction des entités nommées et son contrôle

Toutes les entités nommées détectées au niveau de l'analyse linguistique sont extraites en conservant leur type : « Personne », « Lieu », « Organisme », « Mesure », « Date », « Produit » ou encore « Inconnu ». Mais l'analyse linguistique a pu se tromper sur le type d'une entité nommée. Cette étape effectue un contrôle sur le type des entités nommées extraites.

Le contrôle consiste à vérifier l'adéquation entre le type de l'entité nommée issu de l'analyse linguistique, et le type de l'entité nommée proposé par la règle. Les types des entités nommées peuvent être modifiés si la règle d'extraction considère que le type de l'entité nommée est incompatible avec le type de l'entité nommée issu de l'analyse linguistique.

L'exemple suivant illustre ce phénomène d'incompatibilité : « الجزأير : ُان باريس ُاعلنت *Paris déclare que l'Algérie* ...». Au niveau morpho-syntaxique, Paris est considéré comme une entité nommée de type «lieu», mais dans le cadre de l'action d'émission d'un message, l'agent

ne peut pas être un lieu. En fait, l'émission ne peut être réalisée que par une personne ou une organisation et si à l'origine elle a été considérée comme la capitale de la France, la nouvelle catégorie est une organisation, et nous pouvons en déduire que c'est le gouvernement français.

4.2.3 La création d'entités

Il peut arriver qu'une règle fasse référence à une entité sans que celle-ci existe. C'est le cas lorsque l'entité nommée n'a pas pu être repérée au niveau de l'analyse syntaxique, ou bien lorsqu'il ne s'agit pas d'une entité nommée comme dans l'exemple «أدين : il a été condamné». Pour faire apparaître le patient de l'action, la règle d'extraction va créer l'entité manquante :

*VC#*أَدَانَ *#***<$pers>**# → *<en entype="pers">***<$pers>**</en>

Notons que l'extraction et/ou la création d'entités peuvent introduire des ambiguïtés. Une relation peut demander comme objet ou sujet une entité de type lieu ou organisation. Ces ambiguïtés sont alors générées, pour être résolues plus tard, grâce à un contrôle manuel par exemple, ou bien grâce à la mise en cohérence. Cependant, lorsque l'ambiguïté se situe entre les types personne ou organisation, le type « agent », qui est générique, sera indiqué.

5 Mise en cohérence

Précisons pour commencer que les résultats de l'extraction de connaissances est un graphe **RDF** faisant référence aux concepts et propriétés issus de l'ontologie intégrée dans le système. Cette étape va permettre de rassembler les informations concernant une même entité nommée, ou une même action. Elle permet aussi d'exploiter les métadonnées attachées au document. La construction de notre système d'extraction a nécessité la définition des informations d'intérêt dans le domaine de sécurité.

Nous avons choisi, pour cela, de développer une ontologie du domaine qui servira de guide aux différentes étapes d'extraction. La diversité des documents exploités nécessite que l'ontologie soit assez générale tout en contenant des concepts et des propriétés spécifiques au domaine de la sécurité.

Nous avons développé une ontologie interne, en nous basant sur des ontologies existantes telles que ***foaf*** pour les agents (Person et Organization) et en ajoutant d'autres concepts décrivant les connaissances que nous souhaitons extraire tels que les actes de violences, les déplacements, les transferts d'argent… La construction de cette ontologie a été réalisée manuellement à base de corpus. A l'heure actuelle, notre ontologie compte 106 classes et 200 propriétés d'objets.

5.1 Regroupement des entités nommées

L'un des problèmes des différentes étapes d'extractions réside dans le fait que les graphes obtenus peuvent contenir des duplications inutiles de nœuds. Ce phénomène est particulièrement visible pour les entités nommées que l'on retrouve à plusieurs reprises dans un même document. L'objectif de cette étape consiste à regrouper les différentes occurrences d'une même entité nommée sous un même et unique URI. Ce problème est généralement connu sous le nom de 'Record linkage' ou 'Entity resolution' et a été abordé par différentes approches (Elmagarmid et al., 2007).

Dans le contexte d'un graphe RDF, et dans le domaine de l'extraction sémantique, nous adoptons une méthode basée sur un ensemble de règles. Ces règles ont été définies pour identifier les entités nommées dupliquées et permettre leur regroupement. Citons un exemple de ces règles : deux personnes sont identiques dans un même document, si elles ont le même nom et prénom, et qu'il n'y a pas d'autres informations contradictoires, par exemple «junior» et «senior».

5.2 Résolution des dates relatives

Parmi les problèmes que l'analyse linguistique ne résout pas il y a les dates relatives. Ces dates ne sont pas toujours exprimées d'une manière explicite dans les textes. Pour résoudre ce phénomène, nous nous appuyons sur les trois aspects suivants :

1. La représentation adoptée par l'ontologie : l'ontologie décrit chaque date comme un intervalle. Elle contient donc les attributs suivant : (1) **dtstart :** date de début, (2) **dtend :** date de fin, (3) **type :** le type de calendrier utilisé, qui correspond à des constantes prédéfinies dans l'ontologie (grégorien, arabe, chinois …), (4) **authorValidation :** donne une indication sur quand a eu lieu l'action, si cette dernière se situe dans le passé ou le futur, grâce notamment aux temps des verbes liés à la date, (5) **day :** le jour de la semaine, lorsqu'il est précisé.
2. la sortie de l'analyse linguistique
3. les métadonnées du document analysé (notamment la date d'édition du document).

La sortie de l'analyse linguistique nous permet d'identifier les occurrences où la date extraite est incertaine. Dans le contexte de la presse écrite, il est fréquent d'extraire des dates relatives à un jour de la semaine ou à une indication dans le temps. Par exemple un événement devant se dérouler «نهاية الأسبوع المقبل» : le week-end prochain » pour un article paru le lundi 01 avril 2013 (une métadonnée du document). Les métadonnées sont alors exploitées pour définir une date incertaine se situant entre le samedi 06/4/2013 et le dimanche 07/4/2013.

6 Premiers résultats

Pour estimer l'efficacité de notre système, nous avons mené deux types d'évaluation : une évaluation quantitative concernant la phase de segmentation et la phase d'extraction d'entités nommées et une évaluation qualitative (intrinsèque) de l'extraction de connaissances.

Nous avons comparé nos performances de segmentation avec l'outil de Stanford[2] en apportant quelques modifications aux résultats pour pouvoir les comparer avec ceux de notre outil. Par exemple, dans l'outil de Stanford, l'article défini fait partie du mot, contrairement à notre segmenteur qui considère que l'article défini est un token indépendant.

Nous avons calculé la précision sur un ensemble de documents (articles de presse Aljazeera) segmentés par l'outil de Stanford et corrigés manuellement. Nous avons eu une précision de 0,98% avec notre segmenteur contre 0,96% avec l'outil de Stanford.

[2] http://nlp.stanford.edu/projects/arabic.shtml

Pour évaluer notre approche d'extraction d'entités nommées nous avons réalisé nos expériences sur le corpus ANER[3] (Benajiba et al., 2007) qui est composé de 150 000 occurrences de mots. Ce corpus distingue les types d'entité nommée suivantes : lieu (**LOC**ation qui représente 30.4% des EN observées), personne (**PERS**on : 39%), organisation (**ORG**anization : 20.6%) et une classe qui regroupe toutes les autres EN, de type « divers » (**MISC**ellaneous : 10%). Nous nous sommes intéressés à la reconnaissance des trois premiers types des entités nommées et avons obtenu une précision de 89,05% pour la détection des entités nommées de type personne, 91% pour les lieux et 83.41% pour les organisations.

Nos systèmes de segmentation et d'extraction des entités nommées obtiennent de bons résultats. Par ailleurs, notre système présente encore quelques faiblesses comme le montre la précision pour les entités de types «organisation».

L'absence ou la non disponibilité des outils et des travaux de référence dans le domaine de l'extraction des connaissances pour le traitement de l'arabe a été un vrai obstacle pour mesurer la performance de notre système, et ne nous permet pas de comparer notre approche avec les autres travaux. C'est la raison pour laquelle nous avons lancé des phases de tests afin d'améliorer et de compléter l'extraction d'informations.

Une question engendrée par cette phase est : sur quel corpus peut-on tester notre module ? Nous avons opté pour les corpus suivants :

- Corpus de textes sur Malika El-Aroud. Ce corpus est assez général et regroupe une grande partie des concepts présents dans notre ontologie.
- Ensemble de corpus propres à chaque concept, composés d'articles journalistiques. Ces corpus ne sont pas généraux mais peuvent permettre d'étudier et d'améliorer en profondeur un type de concept. Les corpus spécifiques sont les suivants : « arrestation », « attentat », « condamnation », « construction », « décès », « divorce », « émission », « mariage », « paiement », « rencontre » et « transfert ».

Afin de recouvrir un maximum de cas tout en améliorant la reconnaissance et l'extraction d'information, l'utilisation conjointe de ces deux types de corpus paraît être la meilleure solution.

Citons l'exemple suivant : وفي مرحلة لاحقة إقترنت أُم عُبيدة بالأخ مُعز غرسلاوي من أصول تونسية وانتقلت معهُ لسويسرا. (À un stade ultérieur Umm Obeyda s'est mariée avec le frère Moez Garsallaoui d'origine tunisienne et elle s'est installée (partie) avec lui en Suisse.)

Pour cet exemple, notre système extrait les connaissances suivantes:

- entités nommées de type Personne : Umm Obeyda et frère Moez Garsallaoui.
- entités nommée de type Lieu : Suisse
- concept «Union» extrait grâce au verbe «إقترن» «se marier», avec deux bénéficiaires : «Umm Obeyda » et « Moez Garsalloui».
- concept « Transfert » est extrait grâce au verbe «انتقل».

Voici la représentation des connaissances extraites, dans notre outil de visualisation :

[3] http://users.dsic.upv.es/~ybenajiba/downloads.html

FIGURE 1 – Représentation des connaissances extraites, dans notre outil de visualisation.

Conclusion

Nous avons décrit dans cet article un système d'extraction des connaissances dans des textes arabes, basé d'une part sur une analyse linguistique profonde, et d'autre part sur une extraction sémantique utilisant une ontologie du domaine. L'évaluation effectuée sur ces premiers travaux nous a permis de déceler globalement la qualité de notre extraction mais aussi de donner naissances à d'autres problématiques à étudier.

Notre approche à base de règles contextuelles atteint l'état de l'art pour l'extraction d'entités nommées en arabe. Notre méthode d'extraction de connaissance montre le caractère indispensable d'une analyse syntaxique profonde dans le repérage de telles informations.

Pour rendre notre système d'extraction plus complet, nous allons étendre l'analyse syntaxique, en y ajoutant la recherche des antécédents des anaphores présentes dans les textes. En effet, si les pronoms, utilisés fréquemment dans les textes pour éviter les répétitions, ne sont pas liés à l'entité à laquelle ils font référence, nous risquons de perdre beaucoup des informations présentes dans les textes. Il faut noter que les limites des systèmes linguistiques et statistiques actuels nous orientent vers une future combinaison de ces approches pour une meilleure extraction. Nos travaux futurs s'orientent vers une extension de notre système à d'autres domaines.

Remerciements

Je tiens à remercier l'Agence Nationale de la Recherche portant la référence ANR-09-CSOSG-08-01, pour son aide qu'elle nous a apportée pour mener à bien ce travail, ainsi que Mme Aurélie Pradelles-Rossi et M Christian Fluhr.

Références

BENAJIBA, Y., et ROSSO, P. (2007). ANERsys 2.0 : Conquering the NER Task for the Arabic Language by Combining the Maximum Entropy with POS-tag Information. *In Proc. Workshop on Natural Language-Independent Engineering, 3rd Indian Int. Conf. on Artificial Intelligence, IICAI-2007,* Pune, India, December 17-19.

BRUN, C., DESSAIGNE, N., EHRMANN, M., GAILLARD, B., GUILLEMIN-LANNE, S., JACQUET, G., KAPLAN, A., KUCHARSKI, M., MIGEOTTE, A., NAKAMURA, T. et VOYATZI, S. (2007). Une expérience de fusion pour l'annotation d'entités nommées. *Actes de TALN 2009 (Traitement automatique des langues naturelles)*, Senlis.

CHERFI, H. (2004). Etude et réalisation d'un système d'extraction de connaissances à partir de textes. *Thèse de doctorat*, novembre 2004, LORIA, Nancy.

DEBILI, F. et ACHOUR, H. (1998). Voyellation automatique de l'arabe. *In Proceedings of the Workshop on Computational Approaches to Semitic Languagues.* Montreal. Canada, pages 42–49.

DEBILI, F. et SOUISSI, E. (1998). Étiquetage grammatical de l'arabe voyellé ou non. *Correspondance de l'IRMC, N°71*. Tunis.

ELMAGARMID, A.K., Ipeirotis, P.G. et VERYKIOS, V.S. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and data Engineering (TKDE).* 19(1) pages 1–16.

KUZNIK, L., GUÉNET, A-L., PERADOTTO, A., et CLAVEL, C. (2010). L'apport des concepts métiers pour la classification des questions ouvertes d'enquête. *Actes de TALN 2010 (Traitement automatique des langues naturelles)*, Montréal, Canada .

MESFAR, S. (2008). Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en arabe standard. *Thèse de doctorat*, novembre 2008.

MIKATI, Z. (2010). Du Data Mining au Sense mining : modèle pour une analyse de la langue arabe, et ses représentations formelles en vue d'une application à des données demandant une haute sécurité. *Thèse de doctorat se*, mai 2010.

SÂADANE, H., ROSSI, A., FLUHR, C. et GUIDÈRE, M. (2012). Transcription of Arabic Names into Latin. *Actes the 6th international conference SETIT 2012 (Sciences of Electronic, technologies of Information and Telecommunications).* March 2012. Sousse, Tunisia.

SAMY, D., MORENO, A. et MA GUIRAO, J. (2005). A proposal for an Arabic named entity tagger leveraging a parrallel corpus. *In Proceedings of International Conference on Recent Advances on Natural Language Processing* RANLP '05. Borovets.

SEMMAR, N., GARA, F. et FLUHR, C. (2005). Linguistic resources and analysis for unvowelled Arabic text processing in information retrieval. In *Actes de 2nd International Conference on Machine Intelligence, ACIDCA-ICMI-2005, Tozur (Tunisia)*, 5-7 Novembre 2005.

SHAALAN, K. et RAZA, H. (2009). NERA : Named entity recognition for arabic. *Journal of the American Society for Information Science and Technology,* 60(9) : pages 1652–1663.

ZAGHOUANI, W., POULIQUEN, B., EBRAHIM, M. et STEINBERGER, R. (2010). Adapting a resource-light highly multilingual named entity recognition system to arabic. *Proceedings of the Seventh conference on International Language ressources and Evaluation (LREC'10),* pages 563–567.

ZITOUNI, I., SORENSEN, J., LUO, X. et FLORIAN, R. (2005). The impact of morphological stemming on Arabic mention detection and coreference resolution. *In Proceedings of Workshop on Computational Approaches to Semitic Languages,* pages 63–70, Ann Arbor, Michigan.

ZMANTAR, Y. et DICHY, J. (2009). L'analyse automatique des mots-outils en arabe.*2ème conférence Internationale – Systèmes d'information et Intelligence Economique 2009.* Hammamet,Tunisia.

Extraction des mots simples du lexique scientifique transdisciplinaire dans les écrits de sciences humaines : une première expérimentation

Sylvain Hatier
LIDILEM, BP 25, 38040 Grenoble Cedex 09
sylvain.hatier@u-grenoble3.fr

RÉSUMÉ

Nous présentons dans cet article les premiers résultats de nos travaux sur l'extraction de mots simples appartenant au lexique scientifique transdisciplinaire sur un corpus analysé morpho-syntaxiquement composé d'articles de recherche en sciences humaines et sociales. La ressource générée sera utilisée lors de l'indexation automatique de textes comme filtre d'exclusion afin d'isoler ce lexique de la terminologie. Nous comparons plusieurs méthodes d'extraction et montrons qu'un premier lexique de mots simples peut être dégagé et que la prise en compte des unités polylexicales ainsi que de la distribution seront nécessaires par la suite afin d'extraire l'ensemble de la phraséologie transdisciplinaire.

ABSTRACT

EXTRACTION OF ACADEMIC LEXICON'S SIMPLE WORDS IN HUMANITIES WRITINGS

This paper presents a first extraction of academic lexicon's simple words in french academic writings in the fields of humanities and social sciences through a corpus study of research articles using morpho-syntactic analysis. This academic lexicon resource will be used for automatic indexing as a stoplist in order to exclude this lexicon from the terminology. We try various extraction methods and show that a first simple words lexicon can be generated but that multiwords expressions and words distribution should be taken into consideration to extract academic phraseology.

MOTS-CLÉS : corpus – écrits scientifiques – lexique - phraséologie

KEYWORDS : corpus – scientific writings – lexicon - phraseology

1 Introduction

Notre but est la constitution d'une ressource lexicale du lexique scientifique transdisciplinaire en procédant à son extraction automatique. Ce travail s'inscrit dans le cadre du projet ANR Contint Termith (Terminologie et Indexation de Textes en sciences Humaines), dont le but principal est l'indexation automatique d'écrits scientifiques en sciences humaines. Cette indexation requiert l'identification de la terminologie afin de lister les termes les plus significatifs pour un document donné, terminologie dont un des principaux critères de reconnaissance est la spécificité (propriété des mots statistiquement sur-représenté dans un corpus en comparaison d'une référence). Or, les écrits scientifiques font une large part à un autre lexique spécifique, le lexique scientifique transdisciplinaire, que l'on peut définir comme le lexique servant à la description et la présentation de l'activité scientifique (Tutin, 2007c). Cela nous amène à identifier ce lexique, afin de diminuer le bruit qu'il provoque lors de l'indexation automatique en l'utilisant comme filtre d'exclusion.

Nous nous limiterons ici à l'extraction des mots simples, en nous restreignant aux catégories syntaxiques des noms et des verbes. Nous testons, à l'instar de (Paquot, 2010), différentes méthodes statistiques et procédons à une évaluation humaine des extractions résultantes pour identifier la plus performante. À terme, ces lexiques constitueront une ressource couvrant l'ensemble de la phraséologie de l'écrit scientifique telles les expressions polylexicales (collocations, expressions figées, etc.), et devront être structurés selon une typologie restant à déterminer (syntaxique, notionnelle, fonctionnelle, rhétorique) afin de permettre des applications telles que la caractérisation automatique de documents, l'aide à la rédaction scientifique (pour natifs ou apprenants), l'identification des segments introducteurs de définition ou dénomination.

Notre travail est ciblé sur les écrits dans les sciences humaines et sociales, ceci pour plusieurs raisons. Le projet Termith, auquel nous sommes associé, a comme objet d'étude les écrits en sciences humaines. La distinction entre lexique scientifique transdisciplinaire et terminologie est plus complexe pour les sciences humaines que pour les sciences exactes, dans la mesure où la frontière inter-lexiques y est davantage indéterminée. Enfin, dans une optique de développement de ressource à but pédagogique, un tel travail s'avère particulièrement utile en sciences humaines et sociales où l'écriture académique se révèle plus complexe.

Après avoir présenté dans un premier temps les caractéristiques de l'écrit scientifique et des lexiques le composant, nous reviendrons sur les travaux traitant de notre objet de recherche. Nous détaillerons par la suite la méthodologie d'extraction avant d'analyser les résultats puis nous conclurons sur les apports et limites de notre procédure.

1.1 Écrits scientifiques et lexiques associés:

1.1.1 L'écrit scientifique

Le genre de l'écrit scientifique est particulièrement normé, homogénéisé. Il est fonction de la communauté de discours dans laquelle s'inscrit le scripteur et à laquelle est adressé le discours (Swales, 1990). Dans l'écrit scientifique sont combinés plusieurs types de

lexique : lexique scientifique transdisciplinaire, lexique « abstrait » général (non spécifique aux écrits scientifiques mais très fréquent en rapport de la langue générale), lexique terminologique (lié à la discipline, non traversant), lexique de la langue générale (défini par exclusion des lexiques précédents). Le lexique scientifique transdisciplinaire fait donc partie d'un continuum de lexiques aux frontières floues, lexiques de langue spécialisée dont l'univocité et la monosémie ne sont qu'apparentes (Bertels, 2007). L'extraction d'un lexique commun aux écrits en sciences humaines et sociales, dont la langue, comme le note (Blumenthal, 2007), est différente de celle des sciences exactes, nous permet de mieux caractériser la production des savoirs. De plus, cette extraction participe à concrétiser l'existence d'une communauté de discours qui donne sens à la notion de « transdisciplinarité ».

1.1.2 Définition du lexique scientifique transdisciplinaire

A la suite de (Tutin, 2007a), nous définissons le lexique scientifique transdisciplinaire (désormais LST) comme le lexique renvoyant au discours sur les objets et les procédures scientifiques. Il est par nature non terminologique, et a pour fonction la désignation des procédures et outils de l'activité scientifique. (Da Sylva, 2010) le décrit comme abstrait et largement transdisciplinaire. Pour (Drouin, 2007), le LST se situe au cœur de l'argumentation et de la structuration du discours et de la pensée scientifique. C'est donc un lexique méta-scientifique et méta-discursif (c'est-à-dire qui prend pour objet le discours lui-même).

Le LST a pour principales propriétés d'être :

- transversal aux différentes disciplines, donc réparti dans différents corpus disciplinaires. Ce critère exclut la terminologie, intra-disciplinaire et thématique.
- spécifique à l'écrit scientifique étudié ici, donc absent ou moins fréquent dans la « langue générale » qui sera représentée dans cette étude par un lexique général du français.

Nous présentons ci-dessous un extrait d'article de recherche en psychologie, pour illustrer les différents lexiques présents dans ce genre d'écrit.

Les segments en **gras** appartiennent au LST ou au lexique abstrait général, les segments soulignés à la terminologie.

> […] l'organisation matricielle a **renforcé** et **multiplié** les situations de coopération directe tout au long du **processus**. Zarifian (1996) **estime** qu'on a assisté à un **changement** de **paradigme** et **identifie** une « version faible » de la coopération qui **prévaut** dans les organisations traditionnelles de la conception. **L'objectif** est d'assurer une bonne coordination du travail.[1]

[1]Françoise Darses « Résolution collective des problèmes de conception », *Le travail humain* 1/2009 (Vol. 72), p. 43-59.

2 Travaux sur le lexique de l'écrit scientifique

Plusieurs travaux ont porté sur un lexique spécifique aux écrits scientifiques, majoritairement en anglais. (Coxhead, 2002), par exemple, dans un but didactique, a extrait, en se basant sur les fréquences, une liste de mots anglais. Pour le français, (Phal, 1971) a analysé ce vocabulaire général d'orientation scientifique sur un corpus de manuels scolaires et d'ouvrages non universitaires concernant les sciences dures. Peu d'études se sont donc intéressées à l'écrit scientifique en français dans le domaine des sciences humaines et sociales.

Les différences de procédure, de méthodologie, entre sciences expérimentales et sciences humaines se retrouvent dans les lexiques. Or, la majorité des travaux ont porté sur les sciences expérimentales, ou sur un mélange de sciences exactes et sciences humaines (Paquot, 2010).

Nous nous situons dans la continuité des travaux de (Tutin, 2007c), (Drouin, 2007) et (Da Sylva, 2010), mais en nous appuyant sur un corpus analysé morpho-syntaxiquement uniquement composé d'articles de recherche et ce, seulement en sciences humaines et sociales.

Plusieurs types de statistiques sont utilisés dans les travaux cités. Nous reprenons en partie la méthodologie de (Drouin, 2007) qui combine au critère de fréquence (fréquence relative dans le corpus d'analyse en comparaison avec un corpus de référence) le critère de répartition par tranche. Ce critère permet de s'assurer de la répartition d'un mot dans l'ensemble du corpus et évite ainsi d'extraire des mots certes fréquents mais limités à une sous-partie du corpus. Contrairement à ces travaux ciblés sur les sciences dures, notre corpus d'analyse se restreint aux sciences humaines et sociales. De plus, nous avons pour but la constitution d'un corpus de référence intégrant des textes littéraires, journalistiques ainsi que des transcriptions de l'oral afin de disposer d'un corpus de référence le plus large possible.

En plus de ces critères statistiques de fréquence et de répartition, nous ajoutons un filtrage des segments répétés (afin de ne pas traiter isolément les mots les constituant), et nous nous basons sur un corpus analysé morpho-syntaxiquement. Nous utilisons les méthodes lexicométriques basées sur les spécificités du lexique examiné par comparaison de fréquences, ce qui mène à l'identification de particularités lexicales, subdivisées en spécificités positives (sur-représentation), négatives (sous-représentation) et banales (scores comparables) à partir desquelles nous pouvons décomposer, contrastivement, les différents lexiques constituant notre corpus.

Ces travaux préliminaires devront être poursuivis en se basant sur un corpus de référence du français à large échelle. Le traitement des éléments polylexicaux du LST sera intégré et une typologie des éléments de notre ressource lexicale devra être effectuée afin de la structurer et de permettre des applications didactiques d'aide à la rédaction d'écrits scientifiques.

3 Méthodologie

3.1 Corpus

Pour garantir une homogénéité maximum, le corpus d'analyse est composé d'articles de revues préalablement sélectionnées, dont la qualité est vérifiée par la notation ERIH et/ou AERES et dont le(s) auteur(s) sont francophones natifs. Nous utilisons une sous-partie (ultérieurement augmentée) du corpus du projet Scientext[2].

Notre corpus d'analyse comporte plus de 3,5 millions de mots et sera étendu pour atteindre les 5 millions. Les textes ont été formatés au format TEI Lite (Burnard, 1995) et analysé avec le logiciel Syntex (Bourigault, 2000). Nous utilisons, comme lexique de comparaison de fréquences, la base de données lexicales *lexique3* (New, 2006) qui intègre des informations de fréquence.

Notre corpus d'analyse est précisément composé de 339 articles et de 3 511 716 mots provenant de dix disciplines des sciences humaines et sociales : anthropologie, économie, géographie, histoire, linguistique, sciences de l'éducation, sciences politiques, sciences de l'information, sociologie, psychologie.

3.2 Extraction automatique

Nous avons précédemment défini le LST comme un lexique fréquent, traversant (donc réparti dans les diverses disciplines) et spécifique aux écrits scientifiques (donc plus fréquent dans ces écrits que dans un corpus de référence). Ces hypothèses sur les propriétés linguistiques du LST peuvent se traduire sous forme de critères statistiques que nous appliquons à notre corpus afin d'extraire les éléments qui nous intéressent. Nous combinons les critères suivants :

1. Répartition : le corpus est découpé en 100 tranches de tailles égales. Les mots extraits doivent apparaître dans un minimum de 50 tranches, et un minimum de 5 disciplines sur les 10 composant le corpus d'analyse.
2. Fréquence et Spécifité : les éléments doivent être sur-représenté par rapport au corpus de référence (spécificité positive + seuil du nombre d'occurrences minimal dans l'ensemble du corpus fixé à 100)
3. non-présence systématique dans un segment répété : nous ôtons à la fréquence d'un mot simple le nombre d'occurrences des segments répétés dans lesquels il intervient pour ne pas intégrer des composantes d'unités lexicales qui n'ont pas d'appartenance autonome au LST (par exemple, *point* apparaît une fois sur deux au sein du polylexical *point de vue* et a donc sa fréquence divisée par 2).
4. non-appartenance à une stop-liste ad hoc permettant d'amoindrir le bruit généré par exemple par les segments en langue étrangère (par exemple l'article anglais *the* lemmatisé en nom français *thé*)

Nous calculons le critère de spécificité selon trois formules (ratio de fréquence, chi-carré, rapport de vraisemblance) afin d'identifier, à l'instar de (Paquot, 2009), la plus adaptée,

[2] http://scientext.msh-alpes.fr

en confrontant les différentes listes de mots extraits. Plusieurs calculs peuvent être envisageables, certains offrent de meilleurs résultats sur les événements rares lorsque d'autres fonctionnent mieux sur les événements fréquents : (Labbé, 2001) pointe par exemple la faible efficacité du calcul de spécificité sur les fréquences basses.

Pour les trois calculs statistiques, nous reprenons les formules décrites par Drouin sur le site de son logiciel *TermoStat*[3].

Le fait de travailler sur un corpus analysé morpho-syntaxiquement nous permet d'une part d'effectuer un regroupement flexionnel et ainsi d'amoindrir la dispersion de fréquence, et d'autre part d'utiliser les relations de dépendances récurrentes afin d'ajouter aux lemmes extraits des informations d'ordre lexico-syntaxique. Ces relations, utilisées à ce jour seulement pour contextualiser les mots lors de l'évaluation, devront être intégrées à terme dans le processus d'extraction pour la désambiguïsation.

La fréquence est utilisée de manière absolue, par le biais de seuils, pour valider la présence minimale d'un candidat-LST à l'intérieur d'une discipline et d'une tranche de corpus dans le corpus d'analyse, et de manière relative lors de la comparaison à la base de données qui fait office de corpus contrastif.

Comme il n'existe pas à ce jour de lexique de référence pour vérifier la validité des mots extraits comme éléments du LST, cette appartenance est jugée dans le cadre d'une évaluation effectuée par trois juges experts (chercheurs en linguistique travaillant sur les écrits scientifiques), dont la tâche est présentée dans la section suivante.

3.3 Évaluation

Nous avons créé deux listes, une de 100 verbes et une de 100 noms, compilant les résultats d'extraction des trois différents calculs, en prenant soin de représenter les tranches hautes, moyennes et basses pour chacun d'eux.

Les juges avaient pour consigne de classer ces 200 candidats-LST monolexicaux dans 3 lexiques :

- LST : lexique scientifique transdisciplinaire et lexique abstrait général
- LT : lexique terminologique
- LG : lexique de la langue générale

Ils devaient classer chaque mot dans un lexique au minimum et dans tous au maximum : ceci pour tenir compte de la difficulté à circonscrire ces lexiques comme le note (Tutin, 2007b).

[3]Université de Montréal. Patrick Drouin
http://olst.ling.umontreal.ca/~drouinp/termostat_web/doc_termostat/doc_termostat.html
[consulté le 22/03/2013]

Candidat	LST	LT	LG	Rel()	Contexte
illustrer_V	+	-	-	exemple_N_SUJ cas_N_SUJ	Cet exemple paradigmatique illustre le choix que nous faisons [...]
synthèse_N	+	-	-	proposer_V_OBJ réaliser_V_OBJ	[...] la seconde propose une synthèse des principales interventions [...]

TABLEAU 1 – Exemple de grille d'évaluation

Dans cet extrait de grille d'évaluation, la première colonne présente le candidat-LST sous sa forme lemmatisée suivie de son étiquette de catégorie syntaxique (*N* pour nom et *V* pour verbe).

Les deux dernières colonnes, *rel()* et *contexte,* ont pour fonction d'aider à la décision en apportant une recontextualisation des candidats-LST.

La colonne *rel()* indique les deux associations lexico-syntactiques les plus fréquentes, c'est à dire les couples mot-relation syntaxique les plus fréquemment reliés au candidat-LST (après filtrage des relations peu informatives sur le contexte, telles celles impliquant un verbe auxiliaire ou un pronom). Y sont détaillés le cooccurrent syntaxique, par son lemme et sa catégorie, suivi du type de la relation *(objet* ou *sujet* par exemple).

La dernière colonne *contexte* permet la visualisation d'une phrase contenant l'association lexico-syntaxique la plus fréquente impliquant le candidat-LST.

4 Résultats

4.1 Analyse des évaluations

Pour les verbes, les 3 juges sont en accord sur 68 des 100 candidats (55 sont validés, 13 sont invalidés comme élément du LST). De plus, 27 verbes sont validés par 2 des 3 juges.

Pour les noms, les 3 juges sont en accord sur 79 des 100 candidats-LST (55 sont validés, 24 sont invalidés). 21 noms sont validés par 2 des 3 juges. Dans le tableau ci-dessous, un + représente la validation d'un juge sur l'appartenance au LST, un - représente la non-validation en tant qu'élément du LST.

Appartenance au LST	+ + +	+ + -	- - -	- - +	Accord à 3
Verbes	55	27	13	5	68
Noms	55	21	24	0	79

TABLEAU 2 – Résultats d'évaluation

Ces faibles pourcentages d'accord s'expliquent par deux principaux facteurs :

- L'objet d'étude, le LST, est un lexique à frontière floue, inscrit dans un continuum de lexiques. Il existe ainsi une grande variabilité quant à sa perception selon le juge.
- Les mots s'étudient en contexte. L'apport des associations lexico-syntaxiques récurrentes et la mise en contexte phrastique ne permettent pas une désambiguïsation systématique des différentes acceptions des candidats-LST.

Cette première observation sur l'évaluation met en lumière la difficulté de circonscrire notre objet d'étude, partie d'un continuum qu'il nous faut discrétiser en vue de l'extraction. L'évaluation doit être complétée par une tâche d''annotation en contexte, ce qui permettrait une évaluation précise du silence occasionné par notre méthode d'extraction.

De plus, les cas problématiques d'évaluation sont le plus souvent liés à des mots au sens vague entrant dans des collocations tel *formuler* dans *formuler une hypothèse*. L'ajout futur d'une phase de traitement des expressions polylexicales permettra d'éviter cet écueil.

La difficulté majeure se situe au niveau de la frontière entre LST et lexique de la langue générale : dans tous les cas où les juges ont validé l'appartenance d'un mot à plusieurs lexiques, les lexiques concernés étaient le LST et le lexique de la langue générale.

4.2 Analyse par méthodes statistiques

Nous comparons ci-dessous les 3 formules statistiques utilisées étudiant plus particulièrement les cas où les trois juges sont en accord sur l'appartenance ou non d'un élément au LST.

Nous observons les rangs d'extraction, parmi les trois calculs utilisés, des différents mots de nos listes (validés ou non), le mot de rang 1 étant considéré comme l'élément le plus caractéristique du LST selon le critère de spécificité positive. Les trois calculs que nous utilisons ne donnent pas d'indice certain sur l'appartenance ou non au LST pour un candidat-LST extrait d'après les critères détaillés dans la partie précédente (fréquence haute et répartition large). Par exemple *revenu* est un candidat-LST invalidé par les 3 juges, et classé 82ème selon le ratio, 87ème selon le rapport de vraisemblance et 79ème selon le chi-carré. Nous ne pouvons cependant pas conclure sur l'appartenance au LST des candidats-LST ayant un rang inférieur ou supérieur à celui de *revenu* : *coût*, candidat-LST invalidé, a un rang entre 7 et 13 selon les calculs, alors que *argument*, candidat-LST validé, a un rang entre 214 et 235.

Des mots à spécificité positive, traversant les disciplines, peuvent donc être éléments du LST (*argument, méthode, expliquer*) ou du lexique de la langue générale ou du lexique terminologique *(coût, mobilisation, multiplier)*. Même si 158 des 200 mots sont validés par au moins 2 juges sur 3, le critère de spécificité ne suffit pas à confirmer ou infirmer l'hypothèse d'appartenance au LST. La prise en compte de la fréquence des segments répétés permet d'éliminer certains composants de ces segments mais ne peut gérer les cas d'expressions polylexicales acceptant des modifieurs et donc sujettes à des variations.

En examinant la répartition des occurrences par discipline, nous pouvons dégager un problème récurrent lié aux acceptions variées que peut recouvrir un mot. Par exemple, le nom *coût* a une fréquence totale (dans les 10 disciplines combinées de notre corpus d'analyse) de 976 occurrences, dont 688 dans le seul sous-corpus d'économie, et entre 10 et 80 dans les 9 autres disciplines. En calculant l'écart-type, ce phénomène de sur-présence dans une discipline serait identifié et nous pourrions alors différencier la fréquence du mot dans son acception générale de sa fréquence dans son acception disciplinaire. D'autres exemples plus complexes peuvent être rencontrés, par exemple le cas de *sujet :* il peut avoir plusieurs sens disciplinaires (*Le sujet de la phrase* en linguistique, *Le sujet de l'expérience* en psychologie) ou un sens transdisciplinaire (*Le sujet de l'article*).

Nous pouvons donc observer qu'aucun calcul n'est suffisant pour valider ou invalider de façon certaine un candidat-LST. La fixation d'un seuil ou d'un rang engendre un silence et un bruit trop importants pour ne tenir compte que de ce critère. Nous privilégierons donc la méthode la plus simple, à savoir le ratio de fréquence, et ajouterons le calcul de l'écart-type afin de diminuer le bruit occasionné par les mots à multiples acceptions dont l'une est sur-représentée dans une discipline.

Malgré les apports certains des techniques lexicométriques, celles-ci ne sont pas suffisantes pour extraire automatiquement les mots simples du LST. Une prise en compte de la distribution est nécessaire, que ce soit en vue de la gestion des éléments polylexicaux ou pour la discrimination des diverses acceptions que peut recouvrir un candidat-LST.

Enfin, l'absence d'un lexique de référence du LST, à grande échelle, nous impose une évaluation humaine et ne permet pas de confronter les résultats d'extractions dans leur totalité (nous avons sélectionné 200 mots parmi plus de 1000 composant nos différentes listes) pour quantifier précisément le bruit et le silence générés.

5 Conclusion et perspectives

Nous avons pu, en combinant méthodes statistiques et informations morpho-syntaxiques, procéder à une première extraction des mots simples du lexique scientifique transdisciplinaire dans les articles de recherche en sciences humaines et sociales.

Ces premiers résultats, intéressants du strict point de vue des mots simples, soulignent l'importance de la prise en compte de leur contexte d'apparition, afin de gérer le cas des expressions polylexicales (pour filtrer les mots simples non autonomes), et pour identifier les cas de polysémie et ainsi discriminer les différentes acceptions, transdisciplinaires ou non. La prise en compte de l'écart-type au niveau de la répartition des fréquences intra-disciplinaires est une piste d'identification de ces phénomènes, la distribution en étant une autre, par exemple à l'aide des cooccurrents de deuxième ordre (Bertels 2012).

Le traitement automatique des expressions polylexicales devra être la prochaine étape, étant donné leur part importante dans le LST (Pecman, 2007).

Les travaux de (Kister, 2012) sur le lien syntaxique récurrent entre lexème scientifique transdisciplinaire et terme sont également une piste pour identifier l'acception transdisciplinaire d'un élément ambigu.

L'étape d'évaluation (que nous effectuerons avec plus de juges) devra être remaniée sur deux principaux points :

- Une plus grande recontextualisation des candidats-LST via un plus grand nombre d'exemples phrastiques et d'associations lexico-syntaxiques récurrentes.
- Un travail d'annotation manuelle sur corpus pour évaluer précisément le silence.

Dans l'optique d'affiner les calculs de spécificité, la constitution d'un corpus contrastif « hybride » de grande échelle, intégrant une partie orale, journalistique, et littéraire, sera nécessaire. Le travail ici présenté utilise pour l'étude contrastive une base de donnée ne permettant pas les comparaisons distributionnelles ou la recherche de segments répétés.

Notre corpus d'analyse suivant le format TEI Lite, les informations de structure textuelle pourront être croisées avec les fréquences pour caractériser les parties textuelles d'un article : quel est le lexique le plus présent dans les résumés, conclusions, en regard par exemple des observations sur la prédominance de la terminologie dans les introductions (Rinck, 2010).

Au niveau théorique, l'extraction aura pour but une analyse fine de la phraséologie (au sens large des combinaisons récurrentes et stabilisées) des écrits en sciences humaines et sociales menant à une description de ce type d'écrit. Par ailleurs, compte tenu de la difficile acquisition du métadiscours propre à la production scientifique, une application didactique d'aide à la rédaction pourrait profiter d'une ressource lexicale structurée du LST.

Remerciement

Nous remercions la région Rhône-Alpes qui finance nos travaux de recherche, le projet ANR-Contint Termith, Olivier Kraif pour son aide précieuse sur les aspects lexicométriques ainsi que les relecteurs pour leurs nombreux conseils.

6 Références

Bertels, A. et Geeraerts, D. (2012). L'importance du recoupement des cooccurrents de deuxième ordre pour étudier la corrélation entre la spécificité et la monosémie. *Actes de JADT 2012*, 135-147.

Blumenthal, P. (2007). Sciences de l'Homme vs sciences exactes: combinatoire des mots dans la vulgarisation scientifique. *Revue française de linguistique appliquée*, *12*(2), 15-28.

Bourigault, D., Fabre, C., Frérot, C., Jacques, M. P., et Ozdowska, S. (2005). Syntex, analyseur syntaxique de corpus. In *Actes des 12èmes journées sur le Traitement Automatique des Langues Naturelles*.

Burnard, L., et Sperberg-McQueen, C. M. (1995). TEI lite: An introduction to text encoding

for interchange (*pp. 23-152*). SURFnet.

Coxhead, A. (2002), The academic word list : a corpus-based Word List for Academic Purposes *in Teaching and Language Corpora (TALC) 2000 Conference Proceedings. Atlanta : Rodopi.*

Da Sylva, L. (2010), Extraction semi-automatique d'un vocabulaire savant de base pour l'indexation automatique , *Actes de TALN 2010, 2010.*

Drouin, P. (2007). Identification automatique du lexique scientifique transdisciplinaire. Revue française de linguistique appliquée, 12(2), 45-64.

Evert, S. (2008). Corpora and collocations. Corpus Linguistics. An International Handbook, 2.

Kister, L., et Jacquey, E. (2012). Relation syntaxiques entre lexique terminologique et transdisciplinaire: analyse en texte intégral. *CMLF 2012*.

Labbé C. et Labbé D. (2001). Que mesure la spécificité du vocabulaire?, *Lexicometria, no 3, 23 p.*

New, B., Pallier C., Ferrand L. et Matos R. (2001) Une base de données lexicales du français contemporain sur internet: LEXIQUE, *L'Année Psychologique, 101, 447-462. http://www.lexique.org* [consulté le 22/03/2013]

Paquot, M. et Bestgen, Y. (2009). Distinctive words in academic writing: A comparison of three statistical tests for keyword extraction. *Language and Computers*, *68*(1), 247-269.

Paquot, M. (2010). Academic vocabulary in learner writing: From extraction to analysis. *Continuum*.

Pecman M. (2004), Phraséologie contrastive anglais-français : analyse et traitement en vue de l'aide à la rédaction scientifique. *Thèse de doctorat. 9 déc. 2004. Dir. Henri Zinglé. Université de Nice-Sophia Antipolis. 467 p.*

Pecman, M. (2007). Approche onomasiologique de la langue scientifique générale. *Revue française de linguistique appliquée, 12(2), 79-96.*

Phal, A. (1971), Vocabulaire général d'orientation scientifique (V.G.O.S) – Part du lexique commun dans l'expression scientifique ». *Paris : Didier, Crédif*

Rinck, F. (2010), L'analyse linguistique des enjeux de connaissance dans le discours scientifique, *Revue d'anthropologie des connaissances* 3/2010 (Vol 4, n° 3), p. 427-450.

Swales, J. (1990). Genre Analysis: English in Academic and Research Settings: *Cambridge Applied Linguistics. Cambridge University Press.*

Tutin, A. (2007a). Modélisation linguistique et annotation des collocations: une application au lexique transdisciplinaire des écrits scientifiques. *Formaliser les langues avec l'ordinateur: actes des sixièmes, Sofia 2003, et septièmes, Tours 2004, journées Intex-Nooj*, *3*, 189.

Tutin, A. (2007b). Autour du lexique et de la phraséologie des écrits scientifiques. *Revue française de linguistique appliquée, 12(2),* pages 5-14.

Tutin, A. (2007c). Traitement sémantique par analyse distributionnelle des noms transdisciplinaires des écrits scientifiques. *Actes de Traitement Automatique des Langues Naturelles (TALN),* pages 283-292.

Tutin, A., Grossmann, F., Falaise, A. et Kraif, O. (2009). Autour du projet Scientext: étude des marques linguistiques du positionnement de l'auteur dans les écrits scientifiques. *Actes des 6es journées de linguistique de corpus*.

Vers une identification automatique du chiasme de mots

Marie Dubremetz
Uppsala universitet, Institutionen för lingvistik och filologi - Box 635 - 751 26 Uppsala, Suède
Université Paris Ouest La Défense - 200, Avenue de la République - 92001 Nanterre, France
marie.dubremetz@lingfil.uu.se

RÉSUMÉ

Cette recherche porte sur le chiasme de mots : figure de style jouant sur la réversion (ex. « Bonnet blanc, blanc bonnet »). Elle place le chiasme dans la problématique de sa reconnaissance automatique : qu'est-ce qui le définit et comment un ordinateur peut le trouver ? Nous apportons une description formelle du phénomène. Puis nous procédons à la constitution d'une liste d'exemples contextualisés qui nous sert au test des hypothèses. Nous montrons ainsi que l'ajout de contraintes formelles (contrôle de la ponctuation et omission des mots vides) pénalise très peu le rappel et augmente significativement la précision de la détection. Nous montrons aussi que la lemmatisation occasionne peu d'erreurs pour le travail d'extraction mais qu'il n'en est pas de même pour la racinisation. Enfin nous mettons en évidence que l'utilisation d'un thésaurus apporte quelques résultats pertinents.

ABSTRACT

Towards an automatic identification of chiasmus of words

This article summarises the study of the rhetorical figure "chiasmus" (e.g : "Quitters never win and winners never quit."). We address the problem of its computational identification. How can a computer identify this automatically ? For this purpose this article will provide a formal description of the phenomenon. First, we put together an annotated text for testing our hypothesis. At the end we demonstrate that the use of stopword lists and the identification of the punctuation improve the precision of the results with very little impact on the recall. We discover also that using lemmatization improves the results but stemming doesn't. Finally we see that a French thesaurus provided us with good results on the most elaborate form of chiasmus.

MOTS-CLÉS : chiasme, rhétorique, antimétabole, figure de style.

KEYWORDS: chiasmus, rhetoric, antimetabole, stylistic device.

1 Introduction

1.1 Situation de la recherche

Élevée au rang de science dans l'antiquité, la rhétorique, ou l'étude des techniques de persuasion au moyen du language, a, peu à peu, été délaissée par les recherches linguistiques modernes (Harris et DiMarco, 2009). Parmi les moyens disponibles en rhétorique pour convaincre, on dispose de ce qu'on appelle les figures de style ou figures de rhétorique. Selon Harris et DiMarco (2009) on peut les diviser en deux catégories : les figures de style reposant sur la sémantique

(métaphore, métonymie, comparaison...), et les autres qui jouent sur la syntaxe, les phonèmes ou tout autre constituant de la langue (rime, paronomase...). Si les premières sont relativement populaires, les autres restent un sujet de recherche très rare dans l'horizon académique du TAL. Cet article contribue à combler ce vide en proposant une méthode de détection d'une de ces figures de style : le chiasme de mots. Grâce à des observations empiriques et à l'ajout de contraintes simples (« StopWords » , analyse des ponctuations) nous pensons pouvoir améliorer la qualité des algorithmes existants. Enfin grâce aux outils que nous fournit le TAL (lemmatiseur, raciniseur, thésaurus) nous pourrons envisager de détecter une plus grande variété de chiasmes.

Après avoir présenté les applications liées à notre objet d'étude nous présenterons une définition et le corpus utilisé. De là nous ferons l'état de l'art des recherches existantes avant de soumettre, comparer et enfin discuter notre méthode de détection.

1.2 Applications : pourquoi la question du chiasme en TAL ?

L'application la plus évidente de ce type de recherche semble, à première vue, l'analyse de discours. Ainsi Gawryjolek (2009) avait déjà mis en pratique son outil de détection des figures portant sur la répétition pour analyser un discours de Barack Obama.

On peut aussi imaginer qu'un outil de détection contribuera, à terme, non pas à analyser le style mais à le générer via l'assistance à la rédaction. En effet les logiciels de traitement de textes nous assistent déjà en suggérant les synonymes. À terme le relevé d'un très grand nombre de chiasmes pourrait permettre la constitution de bases de données des figures de style indexées par mots clefs. Pourquoi alors ne pas imaginer un jour que l'utilisateur pourra écrire tout en se voyant suggérer une figure de style appropriée ? Contrairement à ce qu'on pourrait penser, le chiasme n'est pas qu'une coquetterie de style destinée aux seuls poètes. Il est, au contraire, un procédé rhétorique utilisé dans tout texte argumentatif et ce, même s'il est de nature scientifique. [1]

Enfin, il convient d'évoquer une autre application possible, celle de l'extraction automatique de citations (Bendersky et Smith, 2012). Le chiasme en effet génère souvent d'excellents jeux de mots ce qui valorise le texte d'où il est extrait. Étant donnée la nécessité, aujourd'hui, de traiter un grand nombre d'œuvres écrites, repérer non plus juste les mots clefs mais aussi les parties les plus travaillées d'un texte devient de plus en plus important. L'observation du style apporte une nouvelle dimension qui n'existe pas ou peu dans le traitement automatisé.

À présent que nous avons entr'aperçu l'enjeu comment définir notre objet d'étude ?

2 Définition

Le mot chiasme tire son nom de la lettre χ en référence à la croix qu'elle symbolise. On le définit en effet comme la reprise d'un couple d'éléments en sens inverse :

1. « Très répandu dans les années 70, [le chiasme] a été dénoncé vertement pour la violence qu'il fait à la fonction communicative du langage : « [...]. La recherche du sens c'est le sens de la recherche, etc. Vous pouvez paraître profond avec n'importe quelle banalité». Mais en donnant à penser au lecteur naïf, ce chiasme propositionnel [...], est souvent approprié pour des titres, à la fois par son économie lexicale et par la profondeur apparemment inépuisable des discussions qu'il annonce. Dans *La trouble-fête*, Bernard Andrès épingle le procédé comme typique du jargon universitaire, susceptible d'entraîner considération et subventions. »(Vandendorpe, 1991, p.4)

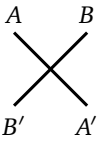


FIGURE 1 – Schéma définitoire du chiasme

On citera ainsi l'exemple le plus classique : « Bonnet blanc, blanc bonnet »

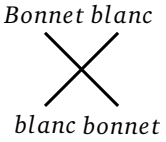


FIGURE 2 – Exemple de chiasme

Il existe toute sorte de chiasmes, ils peuvent porter sur le croisement de phonèmes, de lettres, d'éléments syntaxiques... Nous nous restreindrons à l'étude des chiasmes de mots. Cependant on constate dans cette sous-classe de chiasmes encore beaucoup de variantes qu'il faut définir pour le traitement automatique (Gawryjolek, 2009, p.67). Voici donc, à l'issue des lectures des linguistes [2] et de l'observation des exemples qu'ils donnent, les types de chiasmes de mots que nous avons recensés :

2. Diderot et D'Alembert (1789); Dupriez (2003); García-Page (1991); Greene (2012); Nordahl (1971); Pougeoise (2001); Rabatel (2008); Vandendorpe (1991); Van Gorp *et al.* (2001)

	Nom	Répétition porte sur :	Exemple
Les chiasmes de mots $\begin{matrix} A\ B \\ \times \\ B'\ A' \end{matrix}$	Antimétabole ou Antimétabole stricte	mots identiques [3]	*Ceux qui ont l'opinion que l'argent peut tout faire pourraient bien être suspectés de tout faire pour de l'argent.* (George Savile)
	Chiasme flexionnel ou Antimétabole flexionnelle	mots fléchis	*Lui, veut une alliance des contraires, c'est-à-dire le contraire d'une alliance.* (Le Monde, 13-3-2007)
	Chiasme dérivationnel	mots dérivés	*Moderniser l'Islam plutôt qu'islamiser la modernité.* (Jean Daniel)
	Antimétalepse	idée ou notion sans forcément de reprise morphologique [4]	*dernière averse [. . .] neiges d'antan* (Georges Brassens, *Le temps ne fait rien à l'affaire*)

TABLE 1 – Typologie des chiasmes de mots

Les figures décrites par ce tableau sont très anciennes et connues depuis longtemps. Ainsi, on observe déjà à l'époque de Quintilien (Greene, 2012, Art. « Antimétabole ») l'existence des antimétaboles comportant des mots identiques mais aux flexions différentes. Cependant jamais un nom spécifique n'a été donné à ce phénomène pour le différencier de l'antimétabole « stricte ». Il en va de même pour le chiasme portant sur des mots dérivés. Ce tableau n'est sans doute pas exhaustif puisqu'il est tributaire d'observations empiriques à partir des exemples donnés par les ouvrages de stylistique de référence. Il va cependant permettre à cette recherche d'aller plus loin et de ne pas se heurter au « flottement »(Rabatel, 2008, p.21) qui règne autour de la définition de cette figure : car sans définition formelle, il s'avère difficile d'élaborer un système de détection (Gawryjolek, 2009, p.67).

3. (Dupriez, 2003; Pougeoise, 2001, Art."Chiasme")
4. (Diderot et D'Alembert, 1789, Art. "Antimétabole, Antimétalepse, Antimétathèse")

3 Corpus

3.1 Mode de constitution

Pour évaluer une méthode de détection des chiasmes, l'idéal serait l'obtention d'un corpus déjà annoté. Cependant le chiasme reste un phénomène linguistique relativement rare ce qui rend la constitution d'un corpus difficile et explique en partie l'absence de ressources pour traiter le phénomène. C'est donc grâce aux relevés précis des exemples donnés par les dictionnaires spécialisés (Pougeoise, 2001; Dupriez, 2003), par la base de donnée JULIBEL [5] (Bradfer *et al.*, 1995) par les quelques linguistes ayant traité spécialement de cette figure (García-Page, 1991; Rabatel, 2008; Nordahl, 1971) ou parfois par la lecture d'extraits de textes d'auteurs célèbres tombés dans le domaine public (Hugo, Musset etc.) que nous avons réussi à constituer ce corpus [6]. Le corpus est en langue française si possible non traduite, il est constitué de 43 chiasmes (22 antimétaboles strictes, 8 chiasmes flexionnels, 6 chiasmes dérivationels, 7 antimétalepses). Lorsque nous en avions la possibilité, nous avons ajouté plusieurs pages du contexte d'origine où est apparu le chiasme et vérifié soigneusement que ce contexte ne contenait pas d'autres chiasmes. Concaténés ces chiasmes avec contextes constituent un texte d'entraînement de 16000 mots. Cette taille est déjà assez grande, nous le verrons, pour donner des comparatifs clairs sur l'efficacité des méthodes de détection. Elle reste toutefois assez réduite pour que nous ayons pu vérifier manuellement le nombre précis de chiasmes dans le texte et leur position. Avec toutes ces données des calculs de rappel et de précision seront praticables.

3.2 Contenu

Le corpus sur lequel nous testons est extrêmement varié, à l'image des contextes dans lesquels on peut retrouver cette figure de style. On peut y relever ainsi :

– des textes de littéraires :

> « - Que tu es heureux d'être fou !
> - Que tu es fou de ne pas être heureux ! », (Alfred de Musset, *Les caprices de Marianne*)

– des textes journalistiques :

> « L'économie du discours ne peut se réduire au discours de l'économie. » (*Libération*, paroles rapportées de Michel Debray)

– des textes publicitaires :

> « [A la fin du clip, message au bas de l'écran] "Ne vivez pas pour nettoyer Nettoyez pour vivre " » (RTBF, octobre 2001, clip publicitaire pour *Monsieur Propre*)

Nous mettons à disposition ce corpus sous la forme d'un tableau afin de fournir les métadonnées indispensables à une étude complète. Un chiasme se présentera dans notre tableau sous la forme présentée en table 2.

5. Ressource linguistique destinée à l'enseignement du français `http://home.scarlet.be/lmdp/julibelmoded'emploi.html`, consulté le 08/05/2013

6. Le corpus est mis à libre disposition sur `http://stp.lingfil.uu.se/~marie/chiasme.htm`, consulté le 08/05/2013

Auteur	Citation	Contexte	Source	Chiasme de catégorie :
Cédric Flament	Festival de prix, prix de festival	58e Festival de Cannes * Festival de prix, prix de festival La Face "cachet" de la Croisette	L'Avenir du Luxembourg 20.05.2005 p.14	1

TABLE 2 – Présentation d'un chiasme dans notre tableau

La dernière case est particulière à notre étude. À chaque chiasme nous attribuons sa typologie grâce à un numéro qui correspond aussi au degré de difficulté d'identification en TAL. Voici la légende de cette typologie numérotée (Figure 3).

Catégorie 1 :	Mots strictement identiques, simple repérage de chaînes de caractères identiques
Catégorie 2 :	Mots fléchis, lemmatisation nécessaire
Catégorie 3 :	Mots dérivés d'une même racine, racinisation nécessaire
Catégorie 4 :	Rapprochement sémantique uniquement

FIGURE 3 – Typologie des chiasmes : légende

Nous obtenons ainsi un texte d'entraînement dont les chiasmes sont classés selon le type de traitement nécessaire. Cette classification permet de pratiquer des évaluations spécifiques des algorithmes sur un type de chiasme en particulier.

4 Les méthodes existantes pour la détection des antimétaboles et leur application sur notre corpus

À ce jour il n'existe que deux recherches en TAL qui traitent du phénomène du chiasme, cependant elles ne traitent que de l'antimétabole et s'inspirent du formalisme de Harris et DiMarco (2009) pour définir la figure recherchée (cf. Table 3).

Élément	Signification
w	Mot
...	Suite de caractères
<...>	Frontière de phrase ou de proposition
Indice $_{abc}$	Indique la même identité ou non entre les éléments

TABLE 3 – Formalisme des figures de style [traduit de Harris et DiMarco (2009, p.3)]

Ce formalisme permet à Harris et DiMarco (2009) de définir les antimétaboles ainsi :

$<W_A \ldots W_B \ldots W_B \ldots W_A>$ (Harris et DiMarco, 2009, p.4)

Cette formule signifie qu'une antimétabole est un ensemble de deux couples de mots identiques disposés en inclusion et séparés éventuellement par n'importe quels autres éléments de la langue (autres mots, ponctuations...).

4.1 Algorithme d'identification des doubles paires en inclusion (Gawryjolek, 2009)

En suivant la définition de Harris et DiMarco (2009), Gawryjolek (2009) a mis au point un logiciel qui repère les antimétaboles strictes. Il sélectionne pour cela toutes les doubles paires de mots en inclusion sur une plage spécifiée du texte. Il précise que cette méthode obtient cent pour cent de rappel sur les antimétaboles ce qui, sans surprise, est vérifié sur notre corpus. Il précise aussi qu'il ne filtre aucune paire de mots, même les plus fréquentes, et que la précision est basse sans donner de chiffre ou d'estimation. Nous ne disposons pas du code source de Gawryjolek (2009) ni de la définition de ce qu'il appelle une « plage spécifiée ». Nous avons donc reproduit l'algorithme proposé en limitant la fenêtre de détection à 30 tokens (le plus grand chiasme que nous ayons observé en comportait 23). Concrètement cela signifie que si un chiasme $<W_A \ldots W_B \ldots W'_B \ldots W'_A>$ comporte plus de 30 tokens entre les éléments « W_A » et « W'_A » il est éliminé : c'est une réversion due au hasard sans volonté de provoquer une figure de style, nous appellerons ce phénomène « pseudochiasme ». Dans les conditions que nous avons décrites et sur notre corpus de 16000 mots nous avons obtenu une précision inférieure à 2 % (0.017). Ce manque de précision montre la complexité du problème : une détection efficace des antimétaboles ne peut pas se limiter au repérage de deux couples de mots identiques sans autre forme de sélection.

4.2 Méthode par identification de patrons de trois paires de mots (Hromada, 2011)

Hromada (2011) avec le formalisme de Harris et DiMarco (2009) retient une autre définition de l'antimétabole :

$<W_A\ W_B\ W_C \ldots W_C\ W_B\ W_A>$

Cette formulation signifie qu'une antimétabole est une réversion de trois couples de mots successifs (exemple : « Le pouvoir du discours et le discours du pouvoir) séparés au centre par plusieurs caractères symbolisés par « ... ». Grâce à une expression régulière, il va sélectionner les répétitions non pas de deux mais de trois couples de mots successifs en réversion. Les chiasmes que cette expression régulière sélectionne ont aussi pour contrainte de ne jamais comporter de ponctuation forte. Hromada (2011) ne disposait malheureusement pas d'un corpus manuellement annoté pour évaluer ses résultats. Nous avons donc testé aussi son expression régulière sur notre corpus. Les résultats obtenus sont à l'opposé de ceux de l'algorithme de Gawryjolek (2009) puisque, certes, une antimétabole sur deux n'est pas identifiée (11 sur 22) mais la précision en revanche est parfaite. L'expression recherchée présente donc tant de contraintes qu'elle arrive à ne générer, sur notre corpus, aucun faux positif. Nous n'allons pas reprendre l'idée des trois paires de mots à identifier au lieu de deux car cette mise en œuvre est très restrictive. Cependant

cette méthode se fonde sur une bonne observation : celle que le chiasme en plus des deux paires de mots principaux (comme dans notre exemple les termes « pouvoir » et « discours ») entraîne souvent avec lui la réversion d'autres couples de mots secondaires dans la proposition comme les adverbes ou les déterminants (ainsi le mot « du » dans l'exemple).

5 Méthode proposée

Les algorithmes que nous venons d'étudier et évaluer sont encore assez limités. Grâce aux évaluations précédentes, nous savons que pour détecter les antimétaboles nous avons à notre disposition le choix entre un algorithme exhaustif mais très peu précis (Gawryjolek, 2009) et un algorithme précis mais au rappel beaucoup moins important (Hromada, 2011). L'équilibre entre rappel et précision est en effet difficile à trouver. De plus, dans ces méthodes, la moindre flexion telle qu'un pluriel suffit à ne plus détecter les chiasmes. Nous allons donc tout d'abord mettre au point un algorithme qui concilie rappel et précision avant d'appliquer des outils de TAL pour couvrir d'autres types de chiasmes.

5.1 Les antimétaboles

5.1.1 Méthode

Notre algorithme de détection des antimétaboles s'inspire directement de celui de Gawryjolek (2009). Cependant contrairement à ce dernier, nous prenons en compte les données situées entre les mots répétés. Cela nous permet d'ajouter plusieurs contraintes. Le problème en effet de l'algorithme ne repérant que deux paires de mots en inclusion est qu'il repère bien trop de répétitions de mots non pertinentes (par exemple : « Les chats des villes attrapent des souris mais ne les mangent pas »). Voici donc comment nous procédons pour éliminer ces pseudochiasmes :

1. Nous rejetons les répétitions portant sur les mots les plus courants grâce à une liste de « mots vides » (ou « StopWords »). Nous avons introduit une liste de mots à exclure incluant les déterminants et les mots outils les plus courants (articles, auxiliaires, pronoms et quelques adverbes).
2. Nous éliminons les chiasmes en cas de ponctuation forte à l'intérieur des membre gauche ou droit du chiasme. Nous avons prolongé la réflexion de Hromada (2011) sur les ponctuations. Nous pensons que la ponctuation doit être prise en compte mais contrairement à son expression régulière nous allons tolérer la ponctuation forte dans certains cas. Dans nos exemples en effet on observe que la ponctuation si elle est présente à l'intérieur du chiasme, se trouve le plus souvent, dans la partie centrale (exemple : « Le Parti socialiste est un parti sans leader. François Bayrou est un leader sans parti »). Cela signifie que dans un chiasme $<W_A \ldots W_B \ldots W'_B \ldots W'_A>$ il pourra se trouver un signe de ponctuation forte mais seulement entre les mots W_B et W'_B. Dans le cas contraire, nous considérerons que le chiasme n'est qu'un pseudochiasme. Cette disposition de la ponctuation se justifie par le fait que le chiasme joue souvent sur un effet de symétrie. Ainsi la position centrale du signe de fin de phrase renforce cet effet.
3. Enfin nous conservons la plage de 30 tokens que nous avions utilisée pour l'algorithme de Gawryjolek (2009) (cf. 4.1).

5.1.2 Résultat

Certes cet ajout de contraintes a baissé le rappel par rapport à l'algorithme de Gawryjolek (2009) : sur 22 antimétaboles à retrouver 21 ont été détectées. Le filtre par « StopWords » en effet est une méthode encore trop rudimentaire pour repérer la phrase de Dumas :

> « Tous pour un, un pour tous. »

Cependant le bénéfice sur la précision est très significatif puisque nous passons de 2 à 72 % de précision. Pour trouver 21 chiasmes notre programme ne relève que 36 extraits au lieu de 1235 extraits pour celui de Gawryjolek (2009).

5.2 Les autres types de chiasmes

5.2.1 Les antimétaboles flexionnelles

Pour détecter les chiasmes flexionnels nous avons repris l'algorithme 5.1.1 à la seule différence que nous avons préalablement lemmatisé le texte et les « StopWords » grâce au programme *Treetagger* (Schmid, 1994) associé au lemmatiseur pour le français *Flemm*(Namer, 2000). Nous avons pratiqué le test sur non seulement les antimétaboles flexionnelles mais aussi sur les antimétaboles strictes. Les résultats s'avèrent excellents. 27 antimétaboles sur 30 (22 strictes 8 flexionnelles) ont été trouvées soit un rappel de 90 % et une précision de 46 %. Dans les chiasmes non reconnus on relève :

> « celui qui a le sens de la formule qui ne formule pas beaucoup de sens »

Cette erreur est due non pas à une sous performance du lemmatiseur mais à la nature homonymique du lien entre les deux occurrences du mot « formule ». Ainsi « formule » dans les deux cas a été lemmatisé correctement tantôt sous la forme « formule » (nom) tantôt sous la forme « formuler » (verbe) ce qui empêche l'identification. Dans ce cas précis, la détection avec lemmatisation préalable ne remplace pas la détection sans lemmatisation. À noter, à l'inverse, que les erreurs du lemmatiseur ne gênent pas forcément la détection, ainsi le chiasme :

> « Trop honnête pour être poli que trop poli pour être honnête ! »

lemmatisé par *Flemm* en « Trop honnête pour être **polir** que trop **polir** pour être honnête ! » sera toujours repéré puisque l'erreur provoquée sur « poli » ne change pas l'identité entre les deux occurrences du mot. Ainsi une contre-performance du lemmatiseur n'est pas forcément nuisible à la tâche de détection des chiasmes.

La deuxième erreur de rappel est due à notre filtre par position des ponctuations, l'extrait suivant en effet n'a pas pu être repéré :

> « Faut-il te parler franchement ? ne te riras-tu pas de moi ?
> - Laisse-moi rire de toi, et parle franchement. » Musset, *Les caprices de Marianne*

C'est ici le premier point d'interrogation qui a suscité l'erreur.

Enfin le troisième chiasme non repéré est toujours la devise *des Trois Mousquetaires* (cf. 5.1.2).

5.2.2 Les chiasmes dérivationnels

Pour ce type de chiasme la méthode a été la même que pour son homologue flexionnel à la différence que nous avons ajouté un traitement sur les « StopWords » et sur l'ensemble du texte : la racinisation via le programme *Snowball*[7] (Agichtein et Gravano, 2000). Le résultat pour cet outil est défavorable puisque aucun des six chiasmes nécessitant une racinisation n'a été trouvé. Il y a sur ou sous-racinisation de sorte que les termes du chiasme n'obtiennent jamais la même racine. Est-ce qu'un lemmatiseur à base de dictionnaire et non de règles comme snowball serait plus efficace ? A-t-on vraiment en français les ressources suffisantes pour trouver des chiasmes comme « la régification des stars, la starification des rois » ? On est en présence de jeux morphologiques et étymologiques nécessitant parfois une connaissance académique, ce qu'un raciniseur à base de règle n'a pas.

5.2.3 Les chiasmes jouant sur les liens sémantiques ou antimétalepses

Théoriquement n'importe quel lien sémantique suffit à relier les quatre mots d'un chiasme. En pratique nous avons observé des liens de co-hyponymie :

> « un bâillon pour la bouche et pour la main le clou », Odon Vallet, *L'évangile des païens*

des liens de synonymie ou synonymie partielle

> « Les désespoirs sont morts, et mortes les douleurs », Albert Samain, *Printemps*

et des relations d'antonymie :

> « Ajoutez quelquefois, et souvent effacez. », Nicolas Boileau, *Art poétique*

Nous n'avons pas trouvé de wordnet français couvrant assez de vocabulaire pour travailler sur nos exemples de chiasmes liant des co-hyponymes ou des antonymes. Nous avons toutefois testé une ressource rarement utilisée en TAL (Rao et Ravichandran, 2009, p.677) : le dictionnaire des synonymes *OpenOffice* (OpenOffice-community, 2011).Ce thésaurus, à l'origine utilisé comme dictionnaire de synonymes dans le célèbre traitement de texte présente l'avantage d'être simple (on le trouve sous la forme d'un fichier texte où est inscrit chaque mot associé à une liste de synonymes) traduit dans de nombreuses langues, gratuit et à libre disposition des chercheurs. Comment ce thésaurus permet-il d'identifier des chiasmes sémantiques ? Voici sur un exemple le procédé algorithmique utilisé pour trouver la synonymie. Considérons l'extrait suivant :

> « Dur avec les faibles, et faible avec les forts. »

1. Tout le texte est préalablement lemmatisé. L'extrait devient ainsi : « dur avec le faible , et faible avec le fort . »
2. « dur » est comparé à tous les mots suivants mais il est procédé à chaque fois à deux comparaisons. La première porte sur la morphologie. En effet, lorsque « dur » est comparé au mot « fort » il y a d'abord vérification des deux chaînes de caractères. « dur » n'est pas la même chaîne de caractère que « fort », il faut donc procéder à la seconde comparaison.
3. La seconde comparaison est la comparaison sémantique. Le thésaurus *OpenOffice* est alors mis à contribution. L'ordinateur vérifie l'entrée « dur » du thésaurus, voici un court extrait de ce

7. `http://snowball.tartarus.org/algorithms/french/stemmer.html`, consulté le 08/05/2013

qui y est écrit (cf. Figure 4).

dur|1
(Adverbe Adjectif Nom) | acerbe | acide | consistant | ferme |[...]
| filandreux | nerveux | **fort** | fortin | forteresse | vigoureux |
robuste [...]

FIGURE 4 – Extrait du thésaurus d' *Open Office* à l'entrée du mot "dur"

Le mot « fort » est contenu dans la liste de synonymes du thésaurus donc « dur » et « fort » sont considérés comme les candidats potentiels d'un chiasme sémantique. Si « fort » n'avait pas été contenu dans l'entrée de « dur » la machine aurait aussi vérifié que « dur » n'est pas compris dans l'entrée de « fort » avant d'éliminer les deux candidats.

Nous ne fournirons pas de calcul de précision sur la détection des antimétalepses car le nombre de chiasmes de ce type à trouver étant très réduit (sept) les résultats n'auraient pas de très grande signification. Cependant nous pouvons établir les problèmes rencontrés. Nous avons pu observer une importante baisse de la précision due à des mots outils de sens proches (tant, comme, alors, tel...) ces mots outils étant considérés comme des synonymes par le thésaurus et étant assez fréquents, il nous a fallu les inclure dans les « StopWords » afin d'obtenir des résultats plus clairs. Au final deux antimétalepses ont été repérées avec succès :

« Les désespoirs sont morts et mortes les douleurs »
« Dur avec les faibles, et faible avec les forts »

Nous observons que ces deux chiasmes sont les seuls repérés sur les sept antimétalepses présentes. Cela dit, ce sont aussi les deux seuls reposant sur un lien de synonymie partielle (« Dur/fort », « désespoir/douleur ») nous n'avons donc pas été victime de lacunes en terme de richesse de vocabulaire du thésaurus. Cette ressource n'étant qu'un dictionnaire de synonymes elle ne pouvait pas, par définition, établir un lien entre des antonymes ou des co-hyponymes. En établissant tous les liens synonymiques nécessaires à notre détection, cette ressource a tenu ses promesses.

6 Bilan

Nous avons proposé une méthode permettant d'identifier les chiasmes. Pour ce faire nous avons émis l'hypothèse que par l'ajout de contraintes simples comme le filtrage des mots les plus courants ou la limitation des signes de ponctuation nous pouvions obtenir une meilleure précision. Cette hyothèse est vérifiée. Nous avons ensuite supposé que les outils mis à disposition du TAL français nous aideraient à couvrir une plus grande variété de chiasmes. Cette hypothèse se vérifie largement pour ce qui est du chiasme flexionnel. En revanche, en ce qui concerne le chiasme dérivationnel nous ne sommes pas parvenue avec l'outil de racinisation testé à obtenir de résultats satisfaisants. Enfin l'antimétalepse peut être identifiée automatiquement en français grâce à un thésaurus mais uniquement quand il s'agit de liens de synonymie. Concrétement nous pouvons synthétiser l'apport de notre recherche grâce au tableau ci-dessous (table4).

	Recherche doubles inclusions (Gawryjolek, 2009)	**Recherche triples inclusions (Hromada, 2011)**	**Notre recherche double inclusion avec filtrage + outils TAL**
Précision [8] **(extraits justes / extraits relevés)**	<2 % (22 / 1235)	100 % (11/11)	58 % (21/36)
Rappel (chiasmes relevés / chiasmes à trouver)	100 % (22/22)	50 % (11/22)	95 % (21/22)
F-mesure [9]	4 % (0.035)	66 %	72 %(0.724)
Détecte-t-il les antimétaboles ?	Oui	Oui (en partie)	Oui
Les chiasmes flexionnels ?	Non	Non	Oui
Les chiasmes dérivationnels ?	Non	Non	Non
Les antimétalepses ?	Non	Non	Oui (sur liens de synonymie)
Multilingue	Non testé	Oui	Non
Avantages	– Relevé exhaustif sur les antimétaboles strictes	– Tri manuel des faux positifs quasi-non nécessaire. – Idéal pour fouiller de très grands corpus (>100 000 de mots)	– Relevé des chiasmes jouant sur les mots fléchis – Rappel important – Assez précis pour des analyses semi-manuelles sur corpus de taille moyenne (<100 000 de mots)
Inconvénients	– Peu précis vérification manuelle nécessaire – Sur les corpus >1000 mots tri manuel très long	– Omission de trop d'antimétaboles pour l'analyse de discours	– Tri manuel encore long si corpus très grand (>100 000 de mots). – Exhaustivité non garantie.

TABLE 4 – Synthèse des résultats et comparatif des recherches

8. Tous les chiffres donnés dans ce tableau portent sur la détection d'antimétaboles strictes uniquement : les résultats sur antimétaboles flexionnelles et antimétalepses ne pouvant pas être comparés avec d'autres.

9. On émettra cependant une réserve sur les chiffres au regard de notre corpus (cf. commentaire 7.1).

7 Discussion

7.1 Commentaire sur les résultats

Les résultats du tableau 4 sont à considérer de manière relative. Il est important en effet de rappeler que notre « corpus », utilisé à la fois pour la mise au point et pour l'évaluation de la méthode, même s'il est constitué d'exemples et de contextes réels, ne représente pas la rareté du phénomène du chiasme. Il reste un texte artificiel fondé sur des extraits réels ni plus ni moins. Ainsi, lancé sur un roman d'un 100 000 de mots la précision de notre algorithme comme celle de Gawryjolek (2009) et celle de Hromada (2011) chutera parce qu'il y aura moins de chiasmes à trouver et plus de contexte pour générer des faux positifs.

7.2 Perspectives

À notre connaissance cette recherche est la seule, toutes langues confondues, à proposer une ressource avec des chiasmes classés et dont on connaît précisément l'origine ainsi que le contexte. Cette ressource nécessite bien entendu d'être enrichie surtout en ce qui concerne les antimétalepses et les chiasmes dérivationnels trop rarement illustrés dans les ouvrages de référence. Compte tenu des difficultés de collecte évoquées partie 3.1 et des problèmes de définitions auxquels le chercheur en TAL doit préalablement faire face, une liste d'exemples, même modeste, constitue un vrai tremplin vers l'élaboration d'outils automatiques ou semi-automatiques.

À l'amélioration du corpus, il serait intéressant d'ajouter l'amélioration des algorithmes. Cette recherche nous a déjà permis de faire le pas vers la détection des chiasmes flexionnels : compte tenu des résultats encourageants obtenus avec Flemm, il n'y a plus désormais de raison sur un corpus en français de se contenter de la détection d'antimétaboles strictes comme cela était fait auparavant. Notre recherche introduit aussi la détection des chiasmes ne reposant pas sur le rapprochement morphologique. Enfin les erreurs que nous avons relevées ouvrent la voie vers d'autres réflexions : peut-être que sélectionner manuellement les « stopwords » comme nous l'avons fait n'est pas la manière la plus efficace de procéder. En exploitant d'autres informations comme l'étiquetage grammatical nous espérons lors d'une prochaine étude augmenter la précision voire le rappel. Une analyse linguistique plus poussée non seulement des termes principaux (adjectifs, verbes, noms) mais aussi concernant la disposition des termes secondaires (articles, adverbes, autres mots outils) permettrait peut-être une détection plus pertinente (cf. 4.2).

Remerciements

Merci à mon ancien directeur de recherche le Pr. Marcel Cori de l'université Paris Ouest pour ses conseils avisés lors de la direction de ce travail.

Références

AGICHTEIN, E. et GRAVANO, L. (2000). Snowball : Extracting Relations from Large Plain-Text

Collections. *In Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 85–94, Texas, USA.

BENDERSKY, M. et SMITH, D. (2012). A Dictionary of Wisdom and Wit : Learning to Extract Quotable Phrases. *In Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 69–77, Montréal, Canada.

BRADFER, J., FRANCARD, M. et FAIRON, C. (1995). Base de données Julibel. Centres VALIBEL et CENTAL, `http://julibel.fltr.ucl.ac.be/index.php`, consulté le 08/05/2013.

DIDEROT, D. et D'ALEMBERT, J. l. R. (1789). *Encyclopédie méthodique : ou par ordre de matières, volume 66*. Livre numérique Google `http://books.google.fr/books?id=NchCAAAAYAAJ&lpg=PA198&ots=UEkdoAOE-g&dq=antimetabole%20diderot&pg=PA198#v=onepage&q=antimetabole%20diderot&f=false`, consulté le 08/05/2013.

DUPRIEZ, B. (2003). *Gradus, les procédés littéraires*. Union Générale d'Éditions 10/18.

GARCÍA-PAGE, M. (1991). El "retruécano léxico" y sus límites. *Archivum : Revista de la Facultad de Filología de Oviedo*, 41-42:173–203.

GAWRYJOLEK, J. J. (2009). *Automated Annotation and Visualization of Rhetorical Figures*. Master thesis, Universty of Waterloo.

GREENE, R. (2012). *The Princeton Encyclopedia of Poetry and Poetics : Fourth Edition*. Princeton University Press.

HARRIS, R. et DIMARCO, C. (2009). Constructing a Rhetorical Figuration Ontology. *In Persuasive Technology and Digital Behaviour Intervention Symposium*, pages 47–52, Edinburgh, Scotland.

HROMADA, D. D. (2011). Initial Experiments with Multilingual Extraction of Rhetoric Figures by means of PERL-compatible Regular Expressions. *In Proceedings of the Second Student Research Workshop associated with RANLP 2011*, pages 85–90, Hissar, Bulgaria.

NAMER, F. (2000). Un analyseur flexionnel du français à base de règles. *Traitement Automatique des Langues*, 41:1–23.

NORDAHL, H. (1971). Variantes chiasmiques. Essai de description formelle. *Revue Romane*, 6:219–232.

OPENOFFICE-COMMUNITY (2011). Open Office dictionaries. Thésaurus en français disponible sur : `http://extensions.openoffice.org/en/project/french-dictionary-modern` `http://www.dicollecte.org/download/fr/thesaurus-v2.3.zip`, consultés le 08/05/2013.

POUGEOISE, M. (2001). *Dictionnaire de Rhétorique*. Armand Colin.

RABATEL, A. (2008). Points de vue en confrontation dans les antimétaboles PLUS et MOINS. *Langue française*, 160(4):21–36.

RAO, D. et RAVICHANDRAN, D. (2009). Semi-Supervised Polarity Lexicon Induction. *In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 675–682, Athens, Greece.

SCHMID, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. *In Proceedings of International Conference on New Methods in Language Processing*, pages 44–49, Manchester, Great Britain.

VAN GORP, H., DELABASTITA, D. et D'HULST, L. (2001). *Dictionnaire des termes littéraires*. Honoré Champion.

VANDENDORPE, C. (1991). Lecture et quête de sens. *Protée*, 19:95–101.

Représentation des connaissances du DEC: Concepts fondamentaux du formalisme des Graphes d'Unités

Maxime Lefrançois
WIMMICS, Inria, 2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex
maxime.lefrancois@inria.fr

RÉSUMÉ

Dans cet article nous nous intéressons au choix d'un formalisme de représentation des connaissances qui nous permette de représenter, manipuler, interroger et raisonner sur des connaissances linguistiques du Dictionnaire Explicatif et Combinatoire (DEC) de la Théorie Sens-Texte. Nous montrons que ni les formalismes du web sémantique ni le formalisme des Graphes conceptuels n'est adapté pour cela, et justifions l'introduction d'un nouveau formalisme dit des Graphes d'Unités. Nous introduisons la hiérarchie des Types d'Unités au cœur du formalisme, et présentons les Graphes d'Unités ainsi que la manière dont on peut les utiliser pour représenter certains aspects du DEC.

ABSTRACT

ECD Knowledge Representation : Fundamental Concepts of the Unit Graphs Framework

In this paper we are interested in the choice of a knowledge representation formalism that enables the representation, manipulation, query, and reasoning over linguistic knowledge of the Explanatory and Combinatorial Dictionary (ECD) of the Meaning-Text Theory. We show that neither the semantic web formalisms nor the Conceptual Graphs Formalism suit our needs, and justify the introduction of a new formalism denoted Unit Graphs. We introduce the core of this formalism which is the Unit Types hierarchy, and present Unit Graphs and how one may use them to represent aspects of the ECD.

MOTS-CLÉS : Représentation de Connaissances Linguistiques, Théorie Sens-Texte, Graphes d'Unités, Dictionnaire Explicatif et Combinatoire.

KEYWORDS: Linguistic Knowledge Representation, Meaning-Text Theory, Unit Graphs, Explanatory and Combinatorial Dictionary.

1 Introduction

Dans cet article nous nous intéressons au choix d'un formalisme de représentation des connaissances qui nous permette de représenter, manipuler, interroger et raisonner sur des connaissances linguistiques du Dictionnaire Explicatif et Combinatoire (DEC), qui est le lexique au cœur du sujet d'étude de la Théorie Sens-Texte (TST) (c.f. par exemple Mel'čuk et Arbatchewsky-Jumarie, 1999; Mel'čuk, 2006). Nous envisageons deux scénarios de valorisation d'une telle formalisation :

– Dans un projet orienté vers l'édition lexicographique du DEC, il serait possible de semi-automatiser le travail des lexicographes par exemple en vérifiant qu'un ensemble de contraintes est satisfait, ou en leur suggérant des ébauches d'articles (e.g., liens de fonctions lexicales, ébauche de définition lexicographique, tableaux de régime).
– En proposant une syntaxe basée sur les standards de l'ingénierie des connaissances, les connaissances linguistiques ainsi représentées de manière structurée pourraient être publiées sur le web de données [1] comme l'est aujourd'hui WordNet. Ceci encouragerait leur utilisation comme ressource lexicale hautement structurée par les consommateurs de données du nuage du web de données.

La plupart des projets passés ou présents qui ont consisté en l'informatisation du DEC sont orientés vers l'édition lexicographique. Nous citerons en exemple le projet RELIEF (Lux-Pogodalla et Polguère, 2011) qui vise à représenter un graphe de type système lexical dénommé Réseau Lexical du Français (RLF) (Polguère, 2009) tissé par les liens paradigmatiques et syntagmatiques de fonctions lexicales (e.g., Mel'čuk, 1996). Des travaux de formalisation de certains aspects du DEC ont précédé le projet RELIEF. Citons les travaux de Kahane et Polguère (2001) pour les fonctions lexicales, ainsi que le projet Definiens (Barque et Polguère, 2008) de formalisation des définitions lexicographiques avec genre prochain et différences spécifiques pour le TLFi [2].

En complément de ces travaux de formalisation, notre objectif est de proposer une formalisation au sens de l'ingénierie des connaissances, compatible avec des formalismes standards. Le terme *formalisation* signifie ici non seulement *rendre non-ambigu*, mais également *rendre opérationnel*, i.e., *rendre adapté aux opérations logiques ou rationnelles* (e.g., la manipulation, l'interrogation, et le raisonnement des connaissances). Nous adoptons donc une approche d'ingénierie des connaissances appliquée au domaine de la TST, et la question de recherche de cet article est : *Quel formalisme de représentation des connaissances serait adapté pour représenter les connaissances du DEC ?*

Nous nous intéressons à deux familles de formalismes de représentation des connaissances existant :

– les formalismes du web sémantique, car le web de données est construit dessus ;
– le formalisme des Graphes Conceptuels (GC) (Sowa, 1984; Chein et Mugnier, 2008), puisqu'on sera amenés à faire des raisonnements logiques sur des graphes.

Notre question de recherche se décompose alors en deux sous-questions que nous abordons dans cet article :

– Ces deux formalismes de représentation des connaissances sont-ils adaptés pour représenter les connaissances du DEC ?
– Le cas échéant, comment devons-nous en revoir les bases afin d'en dériver un nouveau formalisme de représentation des connaissances qui soit adapté ?

1. Le web de données est une initiative du W3C en pleine effervescence actuellement, http://linkeddata.org
2. Trésor de la Langue Française informatisé, http://atilf.atilf.fr

La suite de l'article est organisée de la manière suivante. Nous verrons dans un premier temps que ni les formalismes du web sémantique ni les GC ne sont adaptés pour représenter les connaissances du DEC, et nous étayerons le choix suivant : *Nous modifions les bases du formalisme des GC, tout en gardant en tête l'idée d'utiliser les formalismes du web sémantique comme syntaxe pour l'échange des connaissances et pour la publication sur le web de données* (§2). Puisque nous représenterons des unités linguistiques de différentes natures (e.g., sémantème, lexie, grammème, mot-forme), nous choisissons d'utiliser le terme *unité* d'une manière générique et nommons le résultat de cette adaptation *formalisme mathématique des Graphes d'Unités (GU)*. Nous introduirons donc les types unités (§3) puis les graphes d'unités *per se* et leur utilité pour représenter des concepts plus avancés de la TST (§4).

Nous attirons l'attention du lecteur sur le fait que cet article introduit l'élaboration du formalisme mathématique des GU qui fait l'objet d'un rapport de recherche (Lefrançois, 2013). Nous l'invitons à s'y référer pour toute précision définitoire et mathématique.

2 Motivations pour introduire un nouveau formalisme de représentation des connaissances

Les formalismes de représentation de connaissances utilisent abondamment la notion de typage. Les objets du domaine représenté sont nommés instances (ou objets ou individus), et sont typés (ou classifiés). Ils sont liés entre eux par des relations qui sont elles-mêmes typées. Dans cette section nous répondons à la question suivante : *En quoi les formalismes du web sémantique et le formalisme des GC ne sont-ils pas directement adaptés pour représenter les connaissances du DEC ?*

2.1 Les formalismes du web sémantique

On observe un engouement mondial pour les formalismes du web sémantique, et la syntaxe RDF [3] est le standard d'échange de données structurées sur le web de données. L'expressivité de RDF serait suffisante pour représenter les connaissances du DEC. Cependant, la sémantique de RDF, au sens logique, se limite à celle des graphes orientés et étiquetés, et nous souhaitons permettre également de manipuler et de raisonner avec les connaissances linguistiques du DEC. Nous devons donc envisager d'introduire plus de sémantique à l'aide de RDFS [4] ou OWL [5], tout en limitant au maximum le niveau d'expressivité pour conserver de bonnes propriétés computationnelles. OWL introduit de la sémantique à l'aide d'axiomes [6] et de constructeurs de classes et de relations [7]. Justement le projet ULiS (Lefrançois et Gandon, 2011) envisageait une architecture de base de connaissances multilingue compatible avec la TST et basée sur OWL. Dans le projet ULiS, les axiomes et constructeurs de classe de OWL sont utilisés pour que chaque lexie supporte la projection de sa définition lexicographique sur elle-même. Nous avons identifié trois problèmes majeurs avec l'utilisation de OWL pour ce faire :

3. RDF - Resource Description Framework, http://w3.org/RDF/
4. RDFS - RDF Schema, http://www.w3.org/TR/rdf-schema/
5. OWL - Web Ontology Language, http://www.w3.org/TR/owl2-overview/
6. e.g., Sous-classe `SubClassOf( CE1 CE2 )` ; Relation fonctionnelle : `FunctionalObjectProperty( OPE)`
7. e.g., Cardinalité exacte `ObjectExactCardinality( n OPE)` ; Relation inverse `ObjectInverseOf( OPE)`

– Pour chaque définition de lexie on doit introduire autant de nouvelles relations sémantiques qu'il existe de nœuds dans le graphe de définition de la lexie. Cela impose une surcharge de relations superflues ;
– Ces relations doivent être combinées à l'aide de l'axiome de sous-relation d'une relation chaînée `SubObjectPropertyOf( ObjectPropertyChain( OPE`$_1$ `...OPE`$_n$ `) OPE)`, afin de projeter petit à petit le graphe de définition de la lexie sur elle même. Or dans OWL, l'ensemble des relations doit être *régulier* [8] pour garantir la décidabilité des problèmes de raisonnement basiques, et nous avons montré (Lefrançois, 2013) que cette régularité n'est pas assurée dans la petite ontologie donnée en exemple par Lefrançois et Gandon (2011). Cette restriction est donc trop importante pour représenter les définitions du DEC.
– Enfin, la sémantique de l'axiome de sous-relation d'une relation chaînée fait que l'inférence n'est de toute façon possible que dans une direction seulement (sous-relation, et non pas équivalence). C'est à dire que lorsqu'on est en présence du graphe de définition de la lexie, on peut inférer la présence de la lexie, mais pas le contraire.

Une alternative pour représenter les définitions d'unités lexicales serait de les représenter à l'aide de deux règles SPARQL [9] CONSTRUCT réciproques. On se rapporte alors au problème des langages de règles et de leur réconciliation avec OWL (c.f., Krisnadhi *et al.*, 2011), qui ne fait aujourd'hui l'œuvre d'aucun consensus ni standard.

Ces différents problèmes nous poussent à considérer un autre formalisme pour représenter les connaissances du DEC. Nous souhaitons néanmoins un export en RDF pour échanger les connaissances linguistiques sur le web de données.

2.2 Les Graphes Conceptuels

Le formalisme des Graphes Conceptuels (GC) (Sowa, 1984; Chein et Mugnier, 2008) présente de grandes ressemblances avec la TST. Dans leur version basique, les GC représentent des instances typées interconnectées par des relations n-aires également typées. D'ailleurs, l'objectif premier de Sowa était le traitement du langage naturel, et il s'est originellement inspiré des mêmes travaux que les fondateurs de la TST pour mettre au point le modèle des GC : les travaux de Tesnière (1959). Deux des ressemblances les plus marquantes entre les GC et la TST sont les suivantes :
– Dans les GC il est possible de définir des types de concepts et de relations à partir d'un graphe conceptuel, ce qui est très similaire aux définitions des lexies dans la TST ;
– La TST utilise intensivement des règles, en particulier pour les correspondances entre niveaux de représentation d'énoncés. Les règles et leur sémantique, au sens logique, ont été très étudiées dans la littérature des GC.

Un autre atout des GC est le fait qu'il existe des transformations entre les GC et RDF/S (c.f., Corby *et al.*, 2000; Baget *et al.*, 2010). On pourrait donc utiliser ces transformations pour réécrire les GC en RDF pour publication sur le web de données. De plus, pour en revenir au projet ULiS, on pourrait adapter l'architecture du projet ULiS aux GC.

Cependant il n'est pas non plus naturel de représenter les connaissances du DEC à l'aide des GC. Voici deux raisons à cela :
– Un sémantème est modélisable à priori comme un type de concept puisqu'il est instancié dans des représentations sémantiques d'énoncés. D'un autre côté, si la lexie associée est prédicative

8. c.f. par exemple, http://www.w3.org/TR/owl2-syntax#The_Restrictions_on_the_Axiom_Closure
9. SPARQL, http://www.w3.org/TR/sparql11-overview/

et possède des positions actancielles sémantiques, le sémantème peut de manière duale être modélisé par une relation *n*-aire de sorte que ses instances lient d'autres sémantèmes. Les GC ne permettent pas de représenter naturellement cette dualité. En effet, dans les GC on doit respecter une alternance concept/relation, et une représentation sémantique d'énoncé comme celle de la figure 1 ne peut pas être directement représentée par un GC.

– Les positions actancielles sémantiques d'une lexie peuvent différer de celles de la lexie dont son sens dérive [10] (c.f., Mel'čuk, 2004a,b). Or dans les GC, le mécanisme d'héritage des types de relations, qui modélise le fait que *un type de relation est plus spécifique qu'un autre*, est contraint de sorte que deux relations d'arité différente doivent être incomparables. On ne peut donc pas utiliser ce mécanisme naturel d'héritage pour modéliser la spécialisation des sémantèmes.

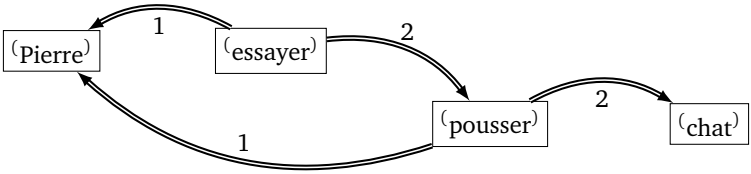


FIGURE 1 – Illustration du rôle dual concept/relation des sémantèmes dans la TST, par la représentation sémantique de *Pierre essaie de pousser le chat*.

2.3 Le nouveau formalisme des Graphes d'Unités

Pour résumer, ni les formalismes du web sémantique ni les GC ne permettent la représentation naturelle des connaissances du DEC. Le formalisme des GC étant le plus proche de la TST, nous décidons donc d'en revisiter les bases afin de le rendre compatible avec la TST.

Puisque nous représenterons des unités linguistiques de différentes nature (e.g., sémantème, lexie, grammème, mot-forme), nous choisissons d'utiliser le terme *unité* d'une manière générique et nommons le résultat de cette adaptation *formalisme mathématique des Graphes d'Unités (GU)*.

Dans un autre travail en cours, nous adaptons les transformations existantes entre les GC et RDF/S (c.f., Corby *et al.*, 2000; Baget *et al.*, 2010) afin d'utiliser les formalismes du web sémantique comme format d'échange des connaissances linguistiques sur le web de données.

Dans la suite de cet article nous apportons une réponse à la question de recherche suivante : *Comment devons-nous revoir les bases du formalisme des GC afin de le rendre adapté à la représentation des connaissances du DEC ?* Cette question se décompose en deux sous-questions :

– Quelle structure mathématique pour une hiérarchie de types d'unités pouvant avoir des positions actancielles (§3) ?
– Quel est l'équivalent des graphes conceptuels pour le formalisme des GU, et comment les utiliser pour formaliser des concepts plus avancés de la TST (§4) ?

10. Par exemple le sémantème (pluie) est plus spécifique que (tomber) mais le sens de *ce qui tombe* et *d'où ça tombe* est figé à (gouttes d'eau) et (ciel/nuage) (Mel'čuk, 2004a).

3 La Hiérarchie des Types d'Unités

Dans cette section nous abordons la question suivante : *Comment devrions-nous revoir les bases du formalisme des GC afin de le rendre adapté à la représentation d'une hiérarchie des types d'unités avec une structure actancielle ?* Tout d'abord, dans le formalisme des Graphes d'Unités (GU), les objets du domaine représenté sont nommés *unités*, et sont typés. A l'instar des formalismes de représentation de connaissances existants et de Mel'čuk (2004a), nous établissons une distinction claire entre :
– Les types d'unités (e.g., type d'unité sémantique, type d'unité lexicale), décrits dans le DEC ;
– Les unités (e.g., unité sémantique, unité lexicale), représentées dans les Graphes d'Unités (GU).
Les types d'unités vont spécifier à travers leurs positions actancielles et signatures comment leurs instances (i.e., unités) devraient être liées entre elles dans un GU. Les types d'unités et leur structure actancielle sont décrites dans une structure dénommée *hiérarchie* et notée $\mathcal{T}$.

3.1 Types d'Unités Primitifs (TPU) et Positions Actancielles

Tout d'abord, $\mathcal{T}$ contient un ensemble fini de Types Primitifs d'Unités (TPU) déclarés noté T_D. Cet ensemble contient des TPU linguistiques de différente nature (e.g., sémantique, lexicale, grammaticale). Afin de nommer les positions actancielles, on introduit un ensemble de relations binaires dénommées *Symboles d'Actants (SymbolA)*, noté $\boldsymbol{S}_{\mathcal{T}}$. $\boldsymbol{S}_{\mathcal{T}}$ contient des numéros pour la structure actancielle des types d'unités sémantiques, et d'autres symboles habituels pour les autres niveaux de représentations considérés (e.g, chiffres romains I à VI pour le niveau syntaxique profond de la TST).

Peut importe qu'il soit sémantique, lexical ou grammatical, un TPU t a un ensemble (qui peut être vide) de *Positions Actancielles (PosA)* dont les symboles sont choisis dans l'ensemble des SymbolA. Certaines PosA peuvent être obligatoires, d'autres optionnelles (Mel'čuk, 2004a), et nous postulons également que certaines PosA peuvent être interdites (i.e., désactivées en quelque sorte). Par exemple le type de lexie TO EAT (⟨manger⟩) a au moins une PosA sémantiques obligatoire qui est l'animal qui mange, et une PosA optionnelle qui est le récipient dans lequel l'animal mange. Si l'on cherche maintenant à affiner le sens de TO EAT pour définir une nouvelle lexie, nous identifions trois cas basiques qui peuvent arriver :
– Une PosA optionnelle peut devenir obligatoire.
– Une PosA optionnelle peut devenir interdite, e.g., le récipient dans TO GRAZE (⟨brouter⟩) ;
– Une nouvelle PosA (à priori optionnelle) peut être introduite ;
Pour représenter ces différents types de PosA et pour que leur présence dans la hiérarchie des types d'unités soit cohérente, on introduit trois fonctions sur l'ensemble des SymbolA :
– $\boldsymbol{\gamma}$ associe à chaque SymbolA $s \in \boldsymbol{S}_{\mathcal{T}}$ son *radix* [11] $\boldsymbol{\gamma}(s)$ qui introduit une PosA de symbole s. On note $\boldsymbol{\Gamma}$ l'ensemble d'arrivée de la fonction $\boldsymbol{\gamma}$, i.e., l'ensemble des *radices* [12].
– $\boldsymbol{\gamma_1}$ associe à chaque SymbolA s son *obligat* [13] $\boldsymbol{\gamma_1}(s)$ qui rend la PosA de symbole s obligatoire. On note $\boldsymbol{\Gamma_1}$ l'ensemble d'arrivée de la fonction $\boldsymbol{\gamma}$, i.e., l'ensemble des *obligant* [14].

11. radix est un mot latin qui signifie ⟨racine⟩.
12. radices est le pluriel de radix.
13. obligat est la forme conjuguée du verbe latin obligo, 3p sing. pres. ind., ⟨il oblige⟩.
14. obligant est la forme conjuguée du verbe latin obligo, 3p plur. pres. ind., ⟨ils obligent⟩.

– $\boldsymbol{\gamma_0}$ associe à chaque SymbolA s son *prohibet* [15] $\boldsymbol{\gamma_0}(s)$ qui rend la PosA de symbole s interdite. On note $\boldsymbol{\Gamma_0}$ l'ensemble d'arrivée de la fonction $\boldsymbol{\gamma_0}$, i.e., l'ensemble des *prohibent* [16].

L'ensemble des *Types Primitifs d'Unités (TPU)* est donc noté $\boldsymbol{T}$ et est égal à l'union disjointe de l'ensemble des TPU déclarés, des radices, des obligant et des prohibent, plus le *TPU universel principal* $\top$ et le *TPU absurde principal* $\bot$.

$$\boldsymbol{T} \stackrel{\text{def}}{=} T_D \uplus \boldsymbol{\Gamma} \uplus \boldsymbol{\Gamma_1} \uplus \boldsymbol{\Gamma_0} \uplus \{\bot\} \uplus \{\top\} \tag{1}$$

On peut alors introduire une relation de spécialisation sur l'ensemble $\boldsymbol{T}$ des TPU sous la forme d'un pré-ordre $\lesssim$. $t_1 \lesssim t_2$ modélise le fait que le TPU t_1 est plus spécifique que le TPU t_2. La relation $\lesssim$ est calculée à partir d'un ensemble de comparaisons déclarées $C_A \subseteq \boldsymbol{T}^2$, et de sorte que :

– $\top$ (resp. $\bot$) soit l'élément maximal (resp. minimal) ;

– pour chaque SymbolA l'obligat et le prohibet soit plus spécifique que le radix.

Chaque PosA ayant un symbole, l'ensemble des PosA d'un TPU $t \in \boldsymbol{T}$ est défini par l'ensemble de leurs symboles $\boldsymbol{\alpha}(t) \subseteq \boldsymbol{S_\mathscr{T}}$. Formellement, l'ensemble $\boldsymbol{\alpha}(t)$ est défini comme l'ensemble des SymbolA dont le radix est plus général ou équivalent à t, et donc tout TPU plus spécifique qu'un SymbolA $s \in \boldsymbol{S_\mathscr{T}}$ hérite d'une PosA de symbole s.

$$\boldsymbol{\alpha}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S_\mathscr{T}} \mid t \lesssim \boldsymbol{\gamma}(s)\} \tag{2}$$

De la même manière, l'ensemble des PosA obligatoires (resp. interdites) d'un TPU t est noté $\boldsymbol{\alpha_1}(t)$ (resp. $\boldsymbol{\alpha_0}(t)$) et est défini comme l'ensemble des SymbolA dont l'obligat (resp. le prohibet) est plus général ou équivalent à t.

$$\boldsymbol{\alpha_1}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S_\mathscr{T}} \mid t \lesssim \boldsymbol{\gamma_1}(s)\} \tag{3}$$

$$\boldsymbol{\alpha_0}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S_\mathscr{T}} \mid t \lesssim \boldsymbol{\gamma_0}(s)\} \tag{4}$$

Finalement, l'ensemble des PosA optionnelles d'un TPU t est noté $\boldsymbol{\alpha_?}(t)$ et est l'ensemble des PosA qui ne sont ni obligatoires ni interdites :

$$\boldsymbol{\alpha_?}(t) \stackrel{\text{def}}{=} \boldsymbol{\alpha}(t) - \boldsymbol{\alpha_1}(t) - \boldsymbol{\alpha_0}(t) \tag{5}$$

Ainsi en descendant la hiérarchie des types d'unités, une PosA de symbole s est introduite par le radix $\boldsymbol{\gamma}(s)$ et définit d'abord une PosA optionnelle pour tout TPU t plus spécifique que $\boldsymbol{\gamma}(s)$, tant que t n'est pas plus spécifique que l'obligat $\boldsymbol{\gamma_1}(s)$ (resp. le prohibet $\boldsymbol{\gamma_0}(s)$) de s auquel cas la PosA devient obligatoire (resp. interdite). La structure actancielle des types d'unités ainsi définie spécifie comment les unités pourront, devront, ou ne devront pas être liées entre elles dans un GU.

15. prohibet est la forme conjuguée du verbe latin prohibeo, 3p sing. pres. ind., ⟨il interdit⟩.
16. prohibent est la forme conjuguée du verbe latin prohibeo, 3p plur. pres. ind., ⟨ils interdisent⟩.

3.2 Signature d'un TPU

Dans les définitions lexicographiques de la TST, le type des unités qui prennent une PosA sémantique est parfois écrit devant le nom de la variable en question. Dans la hiérarchie des types d'unités, les *signatures* des TPU nous permettent de représenter cette information de manière explicite. Plus généralement, les unités qui prennent une certaine PosA d'un TPU doivent avoir un certain type. Par exemple, seules les unités sémantiques peuvent prendre une PosA d'une unité sémantique, et seules les unités de type $^{(}$animal$^{)}$ peuvent prendre la PosA 1 d'une unité de type $^{(}$to eat$^{)}$.

Formellement, l'ensemble des signatures des TPU est noté $\{\boldsymbol{\varsigma}_t\}_{t\in\boldsymbol{T}}$. Pour tout TPU t, $\boldsymbol{\varsigma}_t$ est une fonction qui associe à chaque PosA s de t un ensemble de TPU $\boldsymbol{\varsigma}_t(s)$ qui caractérisent le type des unités qui peuvent prendre cette position. Par exemple la signature de $^{(}$to eat$^{)}$ pour sa PosA 1 est notée $\boldsymbol{\varsigma}_{^{(}\text{to eat}^{)}}(1) = \{^{(}\text{animal}^{)}\}$.

Les signatures participent à la spécialisation de la structure actancielle des TPU, ce qui signifie que si $t_1 \lesssim t_2$ et s est une PosA commune à t_1 et t_2, alors la signature de t_1 pour s doit être plus spécifique que la signature de t_2 pour s. Par exemple, la signature de $^{(}$to sup$^{)}$ pour 1, i.e., $\{^{(}\text{personne}^{)}\}$, est plus spécifique que cette de $^{(}$to eat$^{)}$ qui est $\{^{(}\text{animal}^{)}\}$.

3.3 Hiérarchie des Types d'Unités

Une unité peut en réalité avoir une conjonction, au sens logique, de plusieurs types. En particulier, il peut s'agir d'un type de lexie et de plusieurs types d'unités grammaticales, comme $\{def, plur, \text{CHAT}\}$ pour $^{(}$les chats$^{)}$. Pour représenter ce phénomène, nous introduisons l'ensemble $\boldsymbol{T}^{\cap}$ des *Types Conjonctifs d'Unités (TCU)* possibles sur $\boldsymbol{T}$ comme étant l'ensemble des parties de $\boldsymbol{T}$:

$$\boldsymbol{T}^{\cap} \stackrel{\text{def}}{=} 2^{\boldsymbol{T}} \tag{6}$$

La conjonction des types donne un premier aperçu d'un type d'inférence. En effet, pour une unité de type $\{^{(}\text{personne}^{)}\}$, on peut augmenter son type à $\{^{(}\text{personne}^{)}, ^{(}\text{animal}^{)}\}$ qui est équivalent mais plus "explicite". Plus généralement, un TCU $t^{\cap} \in \boldsymbol{T}^{\cap}$ peut être fermé en y ajoutant tout TPU plus générique qu'au moins un de ses éléments.

Enfin, certains TCU comme $\{def, indef\}$ sont déclarés absurdes, ce qui signifie qu'aucune unité ne peut être à la fois des types def et $indef$. On notera l'ensemble des TCU déclarés absurdes $\bot_A^{\sqcap}$, avec $\bot_A^{\sqcap} \subseteq \boldsymbol{T}^{\cap}$. Par définition, tout type plus spécifique que le TCU absurde principal $\{\bot\}$ est absurde, et pour tout SymbolA $s \in \boldsymbol{S}_{\mathscr{T}}$, le TCU formé de son obligat et son prohibet (i.e., $\{\boldsymbol{\gamma_1}(s), \boldsymbol{\gamma_0}(s)\}$) est absurde.

Nous pouvons maintenant introduire la hiérarchie des types d'unités qui forme le cœur du formalisme des GU. Une hiérarchie de TCU est un n-uplet $\mathscr{T} \stackrel{\text{def}}{=} (T_D, \boldsymbol{S}_{\mathscr{T}}, \boldsymbol{\gamma}, \boldsymbol{\gamma_1}, \boldsymbol{\gamma_0}, C_A, \{\boldsymbol{\varsigma}_t\}_{t\in\boldsymbol{T}}, \bot_A^{\sqcap})$ qui est composé d'un ensemble de Types Primitifs d'Unités déclarés T_D, d'un ensemble de Symboles d'Actants $\boldsymbol{S}_{\mathscr{T}}$, de trois applications $\boldsymbol{\gamma}$, $\boldsymbol{\gamma_1}$ et $\boldsymbol{\gamma_0}$ qui associent à chaque SymbolA ses TPU radix, obligat et prohibet, d'un ensemble de comparaisons déclarées entre TPU C_A, de l'ensemble $\{\boldsymbol{\varsigma}_t\}_{t\in\boldsymbol{T}}$ des signatures des TPU, et d'un ensemble de Types Conjonctifs d'Unités déclarés absurdes $\bot_A^{\sqcap}$,

Les définitions des positions sont étendues aux TCU. L'ensemble des PosA (resp1. PosA obligatoires, resp2. PosA interdites) d'un TCU $t^\cap$ est noté $\boldsymbol{\alpha}^\cap(t^\cap)$ (resp1. $\boldsymbol{\alpha}_1^\cap(t^\cap)$, resp2. $\boldsymbol{\alpha}_0^\cap(t^\cap)$) et est l'union des PosA (resp1. PosA obligatoires, resp2. PosA interdites) des TPU qui le composent. L'ensemble des PosA optionnelles d'un TCU $t^\cap$ est noté $\boldsymbol{\alpha}_?^\cap(t^\cap)$ et est également les PosA qui ne sont ni obligatoires ni interdites. Les signatures sont également naturellement adaptées aux TCU. L'ensemble des signatures des TCU $\{\boldsymbol{\varsigma}_{t^\cap}^\cap\}_{t^\cap \in \boldsymbol{T}^\cap}$ est un ensemble de fonctions de $\boldsymbol{S}_\mathscr{T}$ vers $\boldsymbol{T}^\cap$. Pour chaque TCU $t^\cap$, $\boldsymbol{\varsigma}_{t^\cap}^\cap$ est une fonction avec $domain(\boldsymbol{\varsigma}_{t^\cap}^\cap) = \boldsymbol{\alpha}^\cap(t^\cap)$ qui associe à chaque PosA s de $t^\cap$ l'union des signatures $\boldsymbol{\varsigma}_t(s)$ des TPU t qui composent $t^\cap$.

$$\boldsymbol{\alpha}^\cap(t^\cap) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap} \boldsymbol{\alpha}(t) \tag{7}$$

$$\boldsymbol{\alpha}_1^\cap(t^\cap) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap} \boldsymbol{\alpha}_1(t) \tag{8}$$

$$\boldsymbol{\alpha}_0^\cap(t^\cap) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap} \boldsymbol{\alpha}_0(t) \tag{9}$$

$$\boldsymbol{\alpha}_?^\cap(t^\cap) \stackrel{\text{def}}{=} \boldsymbol{\alpha}^\cap(t^\cap) - \boldsymbol{\alpha}_1^\cap(t^\cap) - \boldsymbol{\alpha}_0^\cap(t^\cap) \tag{10}$$

$$\boldsymbol{\varsigma}_{t^\cap}^\cap(s) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap | s \in \boldsymbol{\alpha}(t)} \boldsymbol{\varsigma}_t(s) \tag{11}$$

Nous avons également introduit une relation de spécialisation sous la forme d'un pré-ordonnancement $\overset{\cap}{\lesssim}$ de l'ensemble des TCU $\boldsymbol{T}^\cap$ tel que (c.f., Lefrançois, 2013) : $\overset{\cap}{\lesssim}$ contient l'extension naturelle d'un pré-ordre sur un ensemble à un pré-ordre sur l'ensemble de ses sous-ensembles ; le bas de $\boldsymbol{T}^\cap$ est aplati de sorte que chaque TCU déclaré absurde soit inférieur à $\{\bot\}$; si la signature d'un TCU pour une PosA est inférieure à $\{\bot\}$, alors ce TCU est inférieur à $\{\bot\}$. Le bas de l'ensemble pré-ordonné $(\boldsymbol{T}^\cap, \overset{\cap}{\lesssim})$ correspond à l'ensemble des TCU équivalents à $\{\bot\}$ et est aplati : il contient $\{\bot\}$, chacun des TCU déclarés absurdes, et plus généralement l'ensemble des TCU qui ne peuvent pas être instanciés. On le nomme ensemble des TCU absurdes.

Nous montrons que les bonnes propriétés des TPU sont préservées par passage aux TCU, à part pour certains cas dégénérés (i.e., type vide et types absurdes).

4 Graphes d'Unités (GU)

Les TCU ainsi définis typeront les unités qui seront représentées par des nœuds unités dans les Graphes d'Unités. Nous allons clore cet article par la définition des GU *per se* et leur utilité pour formaliser des concepts plus avancés de la TST.

4.1 Hiérarchie des Symboles de Circonstants (SymbolC)

Les types d'unités spécifient comment les nœuds unités *sont* liés à d'autres nœuds unités dans un GU. Comme pour tout argument d'un prédicat, une PosA d'une unité ne peut être occupée que par une seule unité à la fois. Cependant on peut également rencontrer des dépendances

d'un autre type dans certaines représentations d'énoncé : des circonstants (Mel'čuk, 2004a). Les relations circonstancielles sont des relations instance-instance contrairement aux relations actancielles qui sont des relations prédicat-argument. C'est le cas des relations syntaxiques profondes non actancielles ATTR, COORD, APPEND par exemple, mais nous pourrons également utiliser ces relations pour représenter le lien entre une lexie et son sémantème par exemple.

Nous introduisons donc un ensemble fini de *Symboles de Circonstants (SymbolC)* noté $\boldsymbol{S}_{\mathscr{C}}$. Pour catégoriser et hiérarchiser l'ensemble des SymbolC, nous introduisons également un pré-ordre $\overset{\mathscr{C}}{\lesssim}$ sur $\boldsymbol{S}_{\mathscr{C}}$ construit par fermeture reflexo-transitive d'un ensemble de comparaisons déclarées $\boldsymbol{C}_{\boldsymbol{S}_{\mathscr{C}}} \subseteq {\boldsymbol{S}_{\mathscr{C}}}^2$. Enfin, à chaque SymbolC est associé une signature qui spécifie de quel type doivent être les unités liées par une relation avec ce symbole. L'ensemble des signatures des SymbolC $\{\boldsymbol{\sigma}_s\}_{s\in\boldsymbol{S}_{\mathscr{C}}}$ est un ensemble de couples de TCU : $\{(domain(s), range(s))\}_{s\in\boldsymbol{S}_{\mathscr{C}}}$. En descendant la hiérarchie des SymbolC et à l'instar des signatures des TCU, nous imposons que la signature d'un SymbolC ne peut que devenir de plus en plus spécifique.

On peut donc introduire la hiérarchie des SymbolC notée $\mathscr{C} \overset{\text{def}}{=} (\boldsymbol{S}_{\mathscr{C}}, \boldsymbol{C}_{\boldsymbol{S}_{\mathscr{C}}}, \mathscr{T}, \{\boldsymbol{\sigma}_s\}_{s\in\boldsymbol{S}_{\mathscr{C}}})$ et composée d'un ensemble fini de Symboles de Circonstants $\boldsymbol{S}_{\mathscr{C}}$, d'un ensemble de comparaisons déclarées de SymbolC $\boldsymbol{C}_{\boldsymbol{S}_{\mathscr{C}}}$, d'une hiérarchie de types d'unités $\mathscr{T}$, et de l'ensemble des signatures des SymbolC $\{\boldsymbol{\sigma}_s\}_{s\in\boldsymbol{S}_{\mathscr{C}}}$.

4.2 Définition des Graphes d'Unités

Les Graphes d'Unités permettent la description d'énoncés à différents niveaux de représentation. A l'instar des GC, les GU sont définis sur un *support* $\mathscr{S} \overset{\text{def}}{=} (\mathscr{T}, \mathscr{C}, \boldsymbol{M})$ qui est composé d'une hiérarchie de types d'unités $\mathscr{T}$, d'une hiérarchie de SymbolC $\mathscr{C}$, et d'un ensemble de marqueurs d'unités $\boldsymbol{M}$. Précisons ce que nous entendons par marqueurs d'unités. Nous établissons une distinction entre :

- les unités, qui sont les objets du domaine représenté ;
- les marqueurs d'unités, qui sont choisis dans l'ensemble $\boldsymbol{M}$, et qui identifient chacun une unité spécifique ;
- les nœuds unité, qui sont interconnectés dans des GU et qui représentent chacun une unité ;
- les marqueurs de nœuds unité, qui sont choisis dans l'ensemble noté $\boldsymbol{M}^{\cap}$ des parties de $\boldsymbol{M}$: $\boldsymbol{M}^{\cap} \overset{\text{def}}{=} 2^{\boldsymbol{M}}$, et qui étiquettent les nœuds unité afin de spécifier quelle unité chaque nœud unité représente.

Ceci peut paraitre complexe au premier abord, mais il s'agit en réalité d'une extension pour la TST qui nous permet d'être proche des GC, et une articulation simple avec les formalismes du web sémantique. En effet, chaque marqueur d'unité correspondra à une URI. Si un nœud unité est étiqueté par le marqueur de nœud unité $\varnothing$ (on dira que c'est un nœud générique d'unité), alors l'unité représentée est inconnue, il sera traduit par un blank-node en RDF. Dans la littérature de la TST, on considère que tous les nœuds des représentations d'énoncés sont génériques. Par contre, si un nœud unité est marqué $\{m_1, m_2\}$, alors les marqueurs d'unité m_1 et m_2 identifient en réalité la même unité, leurs ressources RDF correspondantes seront liées par une relation `owl:sameas`.

Dans leur version simple, les GC possèdent une relation d'équivalence de nœuds concepts nommée *coréférence*. Puisque cette relation ne correspond pas au terme linguistique et que nous représenterons la coréférence linguistique autrement, nous désactivons l'ambiguïté en parlant

plutôt de *relation d'équivalences déclarées de nœuds unités*, notée Eq. Deux nœuds unité déclarés équivalents représentent la même unité. De plus, contrairement à la relation $coref$ des GC, la relation Eq n'est pas une relation d'équivalence sur les nœuds unités [17]. Cela nous permet de distinguer connaissances explicites et connaissances implicites, et facilite l'articulation avec les formalismes du web sémantique.

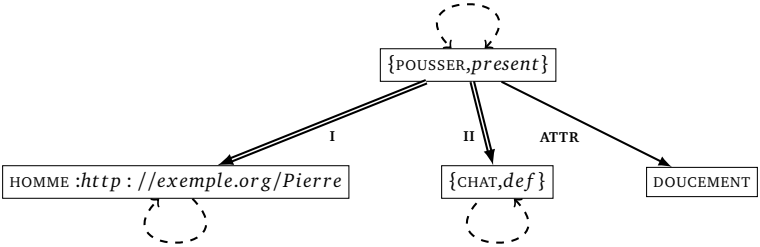


FIGURE 2 – Exemple de Graphe d'Unité : représentation syntaxique profonde de la phrase *Pierre pousse doucement le chat*.

Les GU permettent de représenter des énoncés à différents niveaux de représentation. Nous en avons déjà rencontré un sémantique sur la figure 1, et la figure 2 en représente un syntaxique profond. L'ensemble des GU définis sur un GU-support $\mathscr{S} = (\mathscr{T}, \mathscr{C}, \boldsymbol{M})$ est noté $\mathscr{G}(\mathscr{S})$ et chaque GU $G \in \mathscr{G}(\mathscr{S})$ est un n-uplet $G \stackrel{\text{def}}{=} (U, \boldsymbol{l}, A, C, Eq)$ avec :

– U est l'ensemble des *nœuds unités*. Ils sont illustrés par des rectangles comme sur la figure 2.
– $\boldsymbol{l}$ est une fonction d'étiquetage des noeuds unité. Pour un noeud unité u, $\boldsymbol{l}(u) = (t^\cap, m^\cap)$ est composé d'un TCU $t^\cap \in \boldsymbol{T}^\cap$ qui spécifie la nature de l'unité représentée, et d'un marqueur de nœud unité $m^\cap \in \boldsymbol{M}^\cap$ qui permet d'identifier l'unité représentée le cas échéant. Sur l'exemple de la figure 2, les nœuds unité sont tous typés par des singletons sauf un qui est typé $\{\text{TOMBE},present\}$. De plus, les nœuds unité sont tous génériques sauf pour un marqué $\{http://exemple.org/Pierre\}$.
– A est l'ensemble des *triplets actanciels* $(u, r, v) \in U \times \boldsymbol{S}_{\mathscr{T}} \times U$. Pour tout triplet actanciel (u, r, v), l'unité représentée par v remplit la PosA r de l'unité représentée par u. Ce sont les flèches doubles sur la figure 2.
– C est l'ensemble des *triplets circonstanciels* $(u, r, v) \in U \times \boldsymbol{S}_{\mathscr{C}} \times U$. Pour tout $c = (u, r, v) \in C$, l'unité représentée par u est liée à l'unité représentée par v par une relation circonstancielle de symbole r. Ce sont les flèches simples sur la figure 2.
– $Eq \subseteq U^2$ est l'ensemble d'*équivalences déclarées de nœuds unité*. $(u_1, u_2) \in Eq$ signifie que u_1 et u_2 représentent la même unité. Ce sont les arcs en pointillé sur la figure 2.

4.3 Concepts avancés du formalisme des GU

Les GU sont les briques de base qui vont nous permettre de représenter les connaissance du DEC. Nous allons présenter grossièrement quelques concepts avancés du formalisme des GU.

17. Une relation d'équivalence est une relation réflexive, symétrique et transitive.

Tout d'abord, la définition des GU est permissive et permet par exemple pour un triplet actanciel (u,r,v) d'un GU G, que le type de u n'ait pas de PosA r. A l'instar des formalismes du web sémantique, nous faisons l'hypothèse d'un monde ouvert et considérons que le GU, tout comme la hiérarchie des TCU, représente des connaissances explicites. Nous pouvons donc expliciter dans G le fait que le type de u doive contenir le radix de r. Nous nous inspirons de OWL-RL pour définir l'ensemble des opérations d'explicitation des connaissances, et définissons ainsi la sémantique, au sens logique, des GU. Puisque nous avons dérivé les GC pour définir les GU, nous adapterons les résultats de raisonnement à base de graphes des GC, et définirons la notion d'implication d'un GU par un autre à l'aide d'homomorphismes de graphes. Ces raisonnements logiques seront utiles en particulier pour les représentations sémantiques.

Ensuite, en nous inspirant encore des GC, nous pouvons définir la notion de règles comme un triplet formé d'un GU hypothèse H, d'un GU conclusion C, et d'une bijection κ entre un sous-ensemble de nœuds unités génériques de H et un sous-ensemble de nœuds unités génériques de C. Les règles permettront de représenter les associations sémantème-lexie, et les correspondances entre différents niveaux de représentation (tableaux de régime). Les règles ont été très étudiées pour les GC et nous pourrons adapter aux GU les résultats les concernant.

Nous pouvons également représenter certaines connaissances de la hiérarchie des TCU qui concernent un TPU t dans un GU que l'on nomme *empreinte* de t. L'empreinte de t est un GU avec un nœud central u ayant pour étiquette $\boldsymbol{l}(u) = (\{t\},\varnothing)$, et pour chaque PosA s de t, un autre nœud unité v avec $\boldsymbol{l}(v) = (\varsigma_t(s),\varnothing)$ et un triplet actanciel (u,s,v). Une définition d'une lexie L est alors formée de deux règles réciproques, dont un GU est l'empreinte d'un sémantème t, l'autre GU est la représentation sémantique de la définition lexicographique de la lexie L, et la bijection permet entre autre de repérer le sens du genre prochain de L. A chaque définition correspond donc deux règles d'explicitation des connaissances.

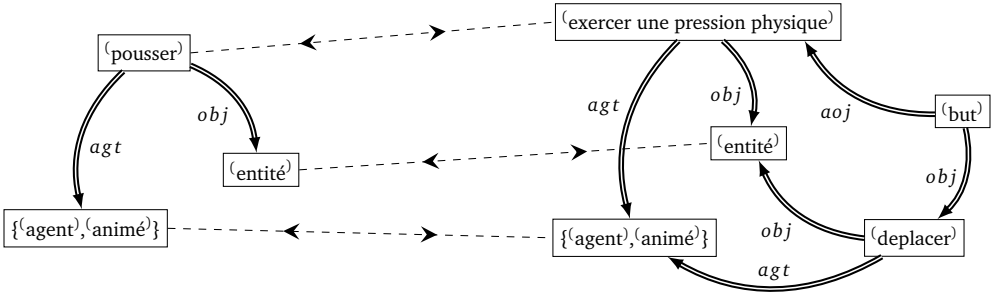


FIGURE 3 – Définition de POUSSER. A gauche, l'empreinte de ⟨pousser⟩.

5 Conclusion

Nous avons donc étudié comment formaliser, au sens de l'ingénierie des connaissances, le Dictionnaire Explicatif et Combinatoire (DEC), et ce afin de pouvoir représenter, manipuler, interroger et raisonner sur des connaissances linguistiques. Nous pouvons maintenant répondre aux questions que nous avons posées.

En quoi les formalismes du web sémantique et le formalisme des Graphes Conceptuels (GC) ne sont-ils pas directement adaptés pour représenter les connaissances du DEC ? La sémantique, au sens logique, de RDF est insuffisante pour représenter les connaissances du DEC, et nous avons montré que l'utilisation de OWL présente des problèmes majeurs. Le formalisme des GC présente des ressemblances avec la Théorie Sens-Texte mais ne permet pas de représenter la dualité concept/relation de la modélisation d'un sémantème. Nous avons proposé de revisiter les bases des GC afin d'en dériver le nouveau formalisme des Graphes d'Unités.

Quelle structure mathématique pour une hiérarchie de types d'unités pouvant avoir des positions actancielles ? Pour prendre en compte la dualité concept/relation des sémantèmes, les relations prédicat-argument sont symbolisées par des Symboles d'Actants (SymbolA), et nous associons à chaque SymbolA trois Types Primitifs d'Unités (TPU) : un radix $\gamma(s)$ qui introduit une Position Actancielle (PosA) de symbol s, un obligat $\gamma_1(s)$ qui rend cette PosA obligatoire, et un prohibet $\gamma_0(s)$ qui rend cette PosA interdite. Ainsi dans l'ensemble pré-ordonné des TPU, une PosA ayant pour SymbolA s est introduite par $\gamma(s)$, et est d'abord optionnelle pour tout TPU plus spécifique que $\gamma(s)$ tant que ce TPU n'est pas plus spécifique que $\gamma_1(s)$ ou $\gamma_0(s)$ auquel cas la PosA devient obligatoire ou interdite. Chaque TPU qui possède des PosA représente donc également un type de relation, qui peut, doit, ou ne doit pas lier une instance de ce type à l'ensemble de ses actants. Enfin, à chaque TPU est associé une signature qui spécifie le type des actants de ses unités. Nous avons étendu les définitions des types d'unités à leur version conjonctive et avons donc introduit la hiérarchie des types d'unités.

Quel est l'équivalent des graphes conceptuels pour le formalisme des Graphes d'Unités (GU), et comment les utiliser pour formaliser des concepts plus avancés de la TST ? Nous avons introduit une hiérarchie des symboles de circonstants. Nous avons ensuite illustré la définition des GU, qui représentent des nœuds unités interconnectés par des relations de dépendance, et des relations d'équivalences déclarée. Nous avons brièvement présenté les concepts plus avancés de la TST que les GU permettent de représenter, et sur lesquels nous travaillons actuellement :

– Nous pouvons définir la sémantique des GU, et donc raisonner avec des représentations d'énoncés.
– Les règles nous permettent de représenter les associations sémantème-lexie, et les correspondances entre différents niveaux de représentation (tableaux de régime).
– Nous pouvons représenter les définitions lexicographiques du DEC à l'aide de deux règles réciproques.

Nous travaillons également sur la factorisation des règles qui nous permettra de représenter des liens de fonctions lexicales, ainsi que sur une syntaxe basée sur les standards du web sémantique pour permettre l'échange standardisé de connaissances du DEC, en particulier sur le web de données.

Remerciements

Je tiens à remercier chaleureusement S. Kahane ainsi que les relecteurs des différentes versions de cet article. Un grand merci également à F. Gandon pour son encadrement, ses précieux conseils et sa disponibilité.

Références

BAGET, J. F., CROITORU, M., GUTIERREZ, A., LECLERE, M. et MUGNIER, M. L. (2010). Translations between RDF (S) and conceptual graphs. *Conceptual Structures : From Information to Intelligence*, page 28–41.

BARQUE, L. et POLGUÈRE, A. (2008). Enrichissement formel des définitions du Trésor de la Langue Française informatisé (TLFi) dans une perspective lexicographique. 22.

CHEIN, M. et MUGNIER, M. L. (2008). *Graph-based Knowledge Representation : Computational Foundations of Conceptual Graphs*. Springer.

CORBY, O., DIENG, R. et HÉBERT, C. (2000). A conceptual graph model for W3C resource description framework. *In* GANTER, B. et MINEAU, G. W., éditeurs : *Conceptual Structures : Logical, Linguistic, and Computational Issues*, numéro 1867 de Lecture Notes in Computer Science, pages 468–482. Springer Berlin Heidelberg.

KAHANE, S. et POLGUÈRE, A. (2001). Formal foundation of lexical functions. *In Proceedings of ACL/EACL 2001 Workshop on Collocation*, page 8–15.

KRISNADHI, A., MAIER, F. et HITZLER, P. (2011). OWL and Rules. *Reasoning Web. Semantic Technologies for the Web of Data*, page 382–415.

LEFRANÇOIS, M. (2013). The Unit Graphs Mathematical Framework. Rapport de recherche RR-8212, INRIA.

LEFRANÇOIS, M. et GANDON, F. (2011). ILexicOn : Toward an ECD-Compliant Interlingual Lexical Ontology Described with Semantic Web Formalisms. *In Proc. of the 5th International Conference on Meaning-Text Theory (MTT 2011)*, page 155–164, Barcelona, Spain. INALCO.

LUX-POGODALLA, V. et POLGUÈRE, A. (2011). Construction of a French Lexical Network : Methodological Issues. *In Proceedings of the International Workshop on Lexical Resources*, Ljubljana.

MEL'ČUK, I. A. (1996). Lexical functions : a tool for the description of lexical relations in a lexicon. *Lexical functions in lexicography and natural language processing*, 31:37–102.

MEL'ČUK, I. A. (2004a). Actants in Semantics and Syntax I : Actants in Semantics. *Linguistics*, 42(1):1–66.

MEL'ČUK, I. A. (2004b). Actants in Semantics and Syntax II : Actants in Syntax. *Linguistics*, 42(2):247–291.

MEL'ČUK, I. A. et ARBATCHEWSKY-JUMARIE, N. (1999). *Dictionnaire explicatif et combinatoire du français contemporain : recherches lexico-sémantiques*, volume 4. PU Montréal.

MEL'ČUK, I. A. (2006). Explanatory Combinatorial Dictionary. *Open problems in linguistics and lexicography*, page 225.

POLGUÈRE, A. (2009). Lexical systems : graph models of natural language lexicons. *Language resources and evaluation*, 43(1):41–55.

SOWA, J. F. (1984). *Conceptual structures : information processing in mind and machine*. System programming series. Addison-Wesley.

TESNIÈRE, L. (1959). *Éléments de syntaxe structurale*. C. Klincksieck (Colombes, Impr. ITE).

Vers un système générique de réécriture de graphes pour l'enrichissement de structures syntaxiques.

Corentin Ribeyre[1, 2]
(1) Université Paris 7 Diderot, 75013 PARIS
(2) INRIA Paris-Rocquencourt, Rocquencourt BP 105 78153 LE CHESNAY
corentin.ribeyre@inria.fr

RÉSUMÉ

Ce travail présente une nouvelle approche pour injecter des dépendances profondes (sujet des verbes à contrôle, partage du sujet en cas d'ellipses, ...) dans un corpus arboré présentant un schéma d'annotation surfacique et projectif. Nous nous appuyons sur un système de réécriture de graphes utilisant des techniques de programmation par contraintes pour produire des règles génériques qui s'appliquent aux phrases du corpus. Par ailleurs, nous testons la généricité des règles en utilisant des sorties de trois analyseurs syntaxiques différents, afin d'évaluer la dégradation exacte de l'application des règles sur des analyses syntaxiques prédites.

ABSTRACT

Towards a generic graph rewriting system to enrich syntactic structures

This work aims to present a new approach for injecting deep dependencies (subject of control verbs, subject sharing in case of ellipsis, ...) into a surfacic and projective treebank. We use a graph rewriting system with constraint programming techniques for producing generic rules which can be easily applied to a treebank. Moreover, we are testing the genericity of our rules by using output of three different parsers to evaluate how the rules behave on predicted parse trees.

MOTS-CLÉS : réécriture de graphes, évaluation de shéma d'annotations, parsing, analyse en syntaxe profonde.

KEYWORDS: graph rewriting system, annotation schemes evaluation, deep syntax parsing.

Introduction

Le French Treebank (FTB) est un corpus arboré du français, qui, comme bien d'autres, se veut neutre d'un point de vue théorique. De fait son schéma d'annotation est à mi-chemin entre les constituants et les dépendances. Le treebank est non configurationnel [1]. C'est pourquoi le schéma d'annotation choisi est assez plat et essentiellement surfacique (Abeillé *et al.*, 2003). On cherche alors à obtenir des représentations plus profondes, telles que le sujet des infinitives, l'antécédent des relatifs, ou encore les sujets elliptiques et le véritable sujet des verbes à contrôle.

Pour ce faire, nous utilisons la réécriture de graphes (Löwe *et al.*, 1993; Geiss *et al.*, 2006) – domaine selon lequel on modifie des graphes au moyen de règles. On recherche une sous-

1. Les structures arborescentes ne sont pas suffisantes pour distinguer un syntagme nominal objet d'un ajout, par exemple.

structure dans le graphe que l'on veut transformer et on applique des modifications à cette structure : ajout d'arcs, de nœuds, modifications des structures de traits sur les nœuds et les arcs, etc. C'est une discipline émergente en traitement automatique des langues, que l'on utilise, par exemple, pour transformer des dépendances surfaciques en graphes sémantiques (Bonfante *et al.*, 2011a). Enfin, la réécriture de graphes peut être mise au profit du passage d'un schéma d'annotation à un autre.

Cependant, lorsque l'on souhaite transformer des structures linguistiques complexes, on se retrouve confronté à deux problèmes : d'une part, la diversité des configurations syntaxiques entraîne une augmentation importante du nombre de règles et, d'autre part, la présence de phénomènes syntaxiques dits « non locaux » demandent des règles complexes, voire récursives, dont l'application est régie par un ordre strict, comme c'est le cas dans le système de réécriture de graphes *Grew* de Bonfante *et al.* (2011b). Par exemple, pour retrouver les antécédents des pronoms relatifs dans le FTB, il faut suivre une chaîne de dépendances plus ou moins longue, afin de remonter du pronom relatif vers l'antécédent. On a aussi les cas de coordinations et notamment de coordinations elliptiques, tels que l'ellipse du sujet où il y a partage du sujet entre les deux conjoints coordonnés. Ajouté à cela le fait qu'il puisse y avoir des constructions elliptiques avec plusieurs modaux entraînant un contrôle, et le partage du sujet devient plus complexe ; c'est notamment le cas dans *Jean pense partir aujourd'hui et rentrer demain*, où la configuration initiale donne *Jean*, sujet de *pense*, or il est aussi le sujet de *partir* et de fait, celui de *rentrer*. En somme, l'écriture de règles pour transformer des graphes peut se révéler longue et difficile, et leur nombre croître rapidement, rendant un tel système difficile à maintenir.

L'approche que nous présentons ici est issue de travaux antérieurs (Ribeyre, 2012; Ribeyre *et al.*, 2012). Nous avons mis en place un système de réécriture qui procède en deux temps : la première technique est fondée sur une approche qui, étant donné un motif de graphe et un graphe de remplacement, tente de retrouver le motif dans un graphe donné et de le remplacer par le graphe de remplacement. Cette approche est largement documentée dans la littérature sur les graphes (Löwe *et al.*, 1993; Geiss *et al.*, 2006). Mais nous apportons une seconde approche, fondée sur la programmation par contrainte (Fruewirth et Abdennadher, 2003), et dite de « propagation de contraintes » sur les arcs. Nous attachons des contraintes sur les arcs du graphe à réécrire et en fonction de celles-ci, des modifications sont effectuées.

Pour valider notre approche, nous avons appliqué une série de règles à deux phénomènes de la syntaxe : (i) le contrôle obligatoire ; (ii) la coordination à ellipse du sujet. Nous cherchons à ajouter les informations manquantes sur un treebank et à tester ensuite leur généricité sur les arbres syntaxiques produits par un analyseur syntaxique (parser) symbolique et deux parseurs statistiques entraînés sur le French Treebank. En effet, les parseurs ne produisant pas toujours d'analyses exactes, nous voulions pouvoir mesurer l'impact des mêmes règles sur des structures syntaxiques plus bruitées. L'idée sous-jacente permet d'ouvrir la voie à la correction d'analyses syntaxiques avec des règles simples.

Nous faisons tout d'abord un rapide état de l'art en matière de réécriture de graphes appliquée à l'enrichissement de corpus, puis présentons notre système de réécriture et plus particulièrement le système de propagation de contraintes, pour ensuite appliquer ce système à la réécriture de phrases issues du FTB (Abeillé *et al.*, 2003) et du SequoiaBank (Candito et Seddah, 2012), phrases qui présentent des phénomènes de contrôle et de coordination à ellipses sujet. Enfin, nous appliquons nos règles aux analyses syntaxiques produites par un parseur symbolique, FrMG (Villemonte de La Clergerie, 2005), et deux parseurs statistiques : le Malt parseur (Nivre

et al., 2006) et le MST parser (McDonald *et al.*, 2005b,a), afin de tester leur généricité. Enfin, nous conclurons sur les perspectives et les autres applications possibles de ce genre de système.

1 Etat de l'art

La réécriture de graphes appliquée à la syntaxe profonde et à l'interface syntaxe-sémantique est un domaine assez jeune. A notre connaissance, seul le système de réécriture *Grew* (Marchand *et al.*, 2010; Bonfante *et al.*, 2010, 2011b) a été utilisé dans ce but.

Grew est un système de réécriture qui s'appuie sur une hiérarchisation en modules pour traiter des phénomènes syntaxiques. Chaque module représente un phénomène bien particulier. De plus, les modules assurent la terminaison du système et une forme de confluence. Cependant, la hiérarchisation des phénomènes syntaxiques en modules est complexe, car les phénomènes interagissent et il devient souvent compliqué de gérer ces interactions, notamment avec les cas de coordinations elliptiques du sujet et de contrôle, mais aussi lors de la recherche de l'antécédent du pronom relatif, comme nous le précisions en introduction. De fait, certaines règles peuvent être présentes dans plusieurs modules, rendant la gestion d'un phénomène parfois éclaté au sein des différents modules.

L'article Bonfante *et al.* (2011b) traite un grand nombre de phénomènes syntaxiques :
– Sujet des participiales, des infinitives et des adjectifs
– Tough movement
– Verbes à contrôle
– Coordinations elliptiques du sujet
– Antécédent des pronoms relatifs
Le système est particulièrement couvrant et le corpus arboré ainsi enrichi se rapproche de la syntaxe profonde. Cependant l'article ne présente aucune évaluation quant à la précision du système et à sa capacité à rendre des analyses correctes.

Nous avons opté pour une évaluation sur des cas complexes et présentant souvent des interactions intéressantes sans pour autant essayer de couvrir un nombre aussi important de phénomènes. Par ailleurs, il nous a semblé tout aussi important de voir à quel point les règles mises au point sur un corpus donné pouvaient encore s'appliquer sur des analyses prédites par des parseurs divers.

2 Présentation du système de réécriture de graphes

OGRE, pour *Optimized Graph Rewriting Engine*, est un système de réécriture de graphes orienté TAL qui permet d'assurer un nombre réduit de règles faciles à maintenir sur des exemples qui peuvent être complexes en terme de réécriture de structures linguistiques. Une description formelle du système peut être trouvée dans (Ribeyre, 2012; Ribeyre *et al.*, 2012), nous rappelons ici les différents types de contraintes et les illustrons sur des exemples linguistiques afin de montrer leurs applications et leurs avantages.

On définit $e = (x \xrightarrow{l} y, C, H)$ un arc étendu qui peut porter un ensemble C potentiellement vide de contraintes et une liste H potentiellement vide représentant un historique formé par des pairs

de nœuds (x', y'). Cet historique nous permet de retracer les changements effectués entre x' et y'. D'autre part, on définit aussi $\mathscr{L}$ qui est l'ensemble des étiquettes possibles sur un arc. On considère alors trois types de contraintes :

– Une contrainte **move up** $m\uparrow$ sur un arc e qui peut être utilisée pour déplacer l'arc e en direction des têtes, comme illustré par la figure 1 [2]. Le déplacement est contrôlé par une paire d'arguments $(\mathscr{A}, q)$ où $\mathscr{A}$ est un automate à états finis et q est un état de $\mathscr{A}$. L'automate représente toutes les transitions possibles à travers lesquelles l'arc peut se déplacer.

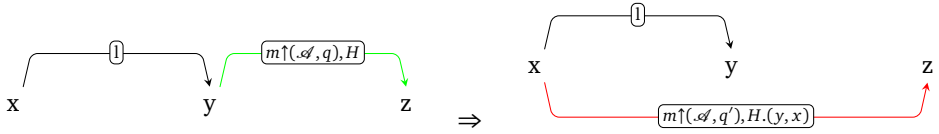


FIGURE 1: Contrainte **move up**

Prenons l'exemple de la figure 2, où l'antécédent du relatif *dont* est *Jean*. Ici, pour retrouver l'antécédent, il faut partir du pronom relatif, et remonter la série d'arcs : *de_obj* → *obj* → *obj* → *obj*, jusqu'à la source de l'arc *mod_rel*. De fait, le nombre d'arcs à remonter n'est pas borné. Il devient donc difficile d'écrire une règle simple pour ce cas de réécriture et il nous faut alors utiliser la contrainte *move up* pour assurer la remontée jusqu'à la source d'un *mod_rel*, assurant que l'on a retrouvé l'antécédent du relatif (l'arc rouge marque la transformation finale).

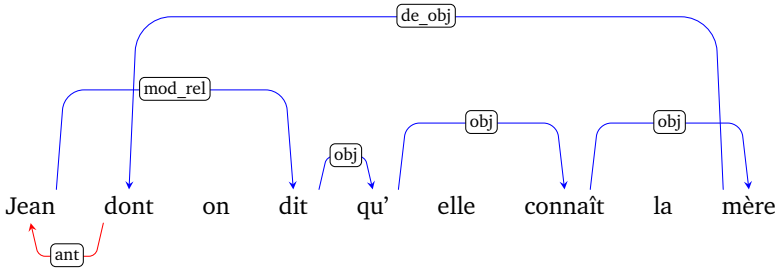


FIGURE 2: Arbre de dépendances pour *Jean dont on dit qu'elle connaît la mère.* En bleu, les arcs à suivre pour retrouver l'antécédent.

– Une contrainte **share up** $s\uparrow$ sur un arc $e = y \xrightarrow{l_e} z$ qui peut être utilisée pour dupliquer tous les arcs entrants $e' = x \xrightarrow{l} y$ de y comme arcs entrants de z (voir figure 3). Comme la contrainte $r\uparrow$, cette contrainte accepte un argument **L** qui restreint la duplication des arcs e' aux seuls arcs avec une étiquette $l \in \mathbf{L}$.

– Une contrainte **share down** $s\downarrow$ sur un arc $e = y \xrightarrow{l_e} z$ peut être utilisée pour dupliquer tous les arcs sortants de y comme arcs sortants de z, (voir figure 4). Cette contrainte accepte un argument **L** qui restreint la duplication des arcs e' avec une étiquette $l \in \mathbf{L}$. A noter, les arcs résultants ont une étiquette l^+, qui indique qu'ils ont été clonés.
Dans la figure 5, on remarque la présence de deux modaux, *vouloir* et *pouvoir*.

2. Où les arcs verts sont supprimés du graphe et les arcs rouges correspondent à la transformation finale.

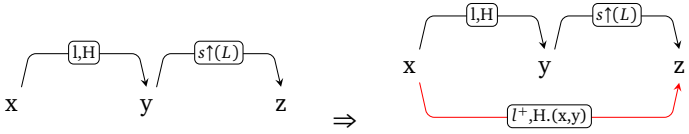


FIGURE 3: Contrainte **share up**

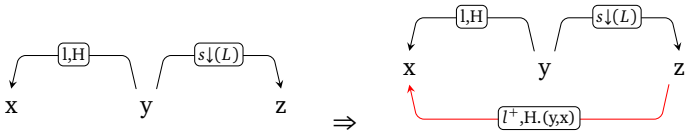


FIGURE 4: Contrainte **share down**

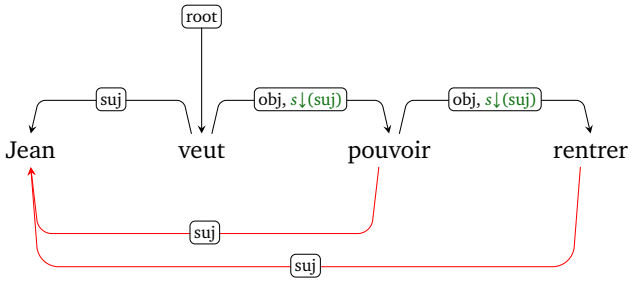


FIGURE 5: Arbre de dépendances pour *Jean veut pouvoir rentrer.*

Poser une contrainte de type *share_down* sur les deux arcs d'étiquette **obj**, conduit au partage du sujet, permettant ainsi de retrouver le sujet des verbes dans des phrases où il y en a plusieurs d'affilés.

Par ailleurs, les contraintes sont conçues pour permettre une interaction qui favorise la confluence et la terminaison du système. Prenons l'exemple *Jean qui m'aperçoit et m'appelle veut me parler*. On donne une représentation de cette phrase à la figure 6. Les arcs rouges décrivent les arcs finaux (après réécriture), l'arc portant la contrainte **move up**, qui est en vert foncé, est susceptible de bouger lors de la transformation et l'arc bleu est un arc temporaire détruit après transformation.

On voit qu'on utilise les trois contraintes précédemment explicitées. Deux contraintes **share down** permettent d'ajouter le sujet de *parler* et celui d'*appelle*. La contrainte **move up** ajoute l'antécédent du relatif « *qui* » ainsi qu'une contrainte **share up** qui permet d'ajouter le véritable sujet des verbes *aperçoit* et *appelle*.

La confluence est assurée par le fait que **move up** ne peut pas utiliser les arcs créés par **share down** ou **share up** et par le fait que les contraintes s'appliquent tant qu'elles le peuvent. Ainsi, on pourrait appliquer **move up**, puis **share down** et **share up** ou inverser **move up** et **share down** sans problème, le résultat serait toujours le même. Enfin, on voit que le nombre d'informations ajoutées est important au regard du nombre de contraintes posées sur les arcs.

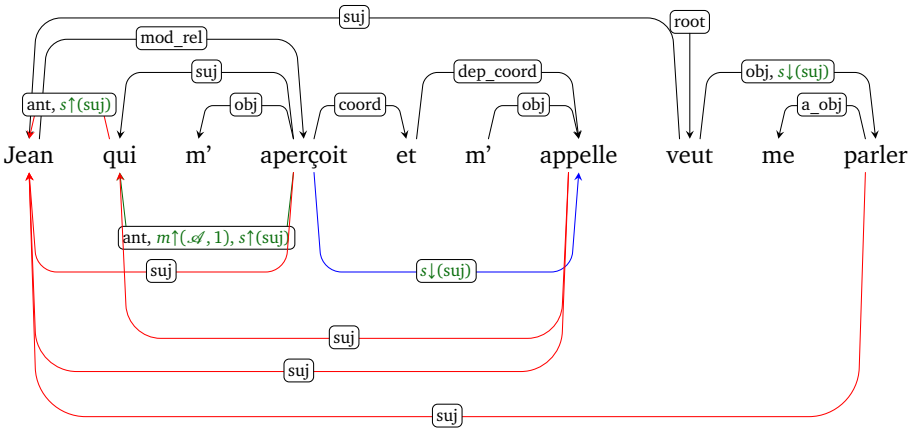


FIGURE 6: Exemple d'interaction entre contraintes

3 Illustration sur le contrôle du sujet et l'ellipse du sujet

Comme expliqué plus haut, nous avons utilisé notre système de réécriture pour ajouter des informations sur les cas de contrôles et d'ellipses du sujet.

3.1 Contrôle du sujet

Le contrôle du sujet est un problème lexical. En effet, la liste des verbes à contrôle est connue et peut être récupérée via un lexique tel que le LEFFF (Sagot, 2010). Cependant, il existe des exemples où le nombre de verbes à contrôle mis en relation peut être important, nous citerons le cas suivant issu du French Treebank : *Le Brésil veut pouvoir continuer à défricher l'Amazonie pour y installer ses colons affamés de terres cultivables* (phrase **5444** dans le FTB, section 1).

Nous sommes en présence d'un exemple qui demande de propager le sujet *Brésil* tout au long de la chaîne de modaux, jusqu'au verbe *défricher*. Ainsi, *Brésil* est sujet de *veut*, mais aussi de *pouvoir* et de *continuer à* et enfin de *défricher*.

3.2 Ellipse du sujet

Nous donnons à la figure 7 un exemple de représentation de l'ellipse sujet dans le French Treebank en dépendances.

Dans cet exemple, il faut partager le sujet *Jean* entre les deux conjoints de la coordination. Placer une contrainte *share down* entre *mange* et *boit* permet ce partage.

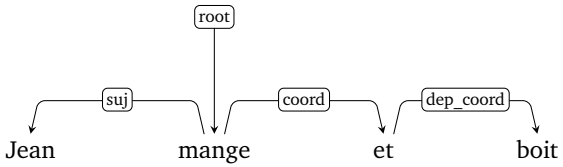


FIGURE 7: Représentation en dépendances de l'ellipse sujet selon le schéma d'annotation du FTB

4 Validation sur corpus

Pour valider notre travail, nous avons mis au point deux corpus, que nous décrivons en détail ci-dessous.

4.1 Les corpus

Pour nous permettre de nous évaluer, nous avons mis en place un sous-corpus par phénomène considéré. Ils sont tirés d'un ensemble de phrases du French Treebank (FTB) et du Sequoia-Bank (Candito et Seddah, 2012).

Ainsi, nous avons sélectionné un corpus de 405 phrases pour le contrôle obligatoire issues du SequoiaBank et du French Treebank que nous avons annotées à la main. En ce qui concerne la coordination à ellipse sujet, nous avons utilisé le corpus annoté de Bouchesèche (2009), auquel nous avons ajouté de nouvelles phrases pour un total de 120 phrases prises sur le French Treebank (section 2 et section 3 uniquement).

La répartition des phrases et du nombre total de dépendances pour chaque corpus est récapitulée dans le tableau 1.

PHÉNOMÈNE	DEV	TEST	TOTAL
Contrôle	155 (4223)	250 (6572)	405 (10795)
Ellipse du sujet	55 (1920)	65 (2423)	120 (4343)

TABLE 1: Nombre total de phrases (et de dépendances) dans les sous-corpus par phénomène étudié

Nos règles ont été mises au point sur les deux corpus de développement pour ensuite être testées sur les corpus de test. L'idée est de ne pas biaiser l'évaluation en les testant sur des phrases déjà connues.

Dans le tableau 2, nous indiquons le nombre de dépendances ajoutées lors de l'annotation manuelle.

Par ailleurs, on tient à attirer l'attention sur le fait que les deux phénomènes ne peuvent être comparés. En effet, la difficulté de la tâche est différente et le nombre de phrases dans chaque

PHÉNOMÈNE	DEV	TEST	TOTAL
Contrôle	345	578	923
Ellipse du sujet	116	129	145

TABLE 2: Nombre de dépendances ajoutées dans chaque sous-corpus

corpus n'est pas le même.

4.2 Protocole expérimental

Parseurs Pour tester la généricité de nos règles, nous avons analysé les corpus avec trois parseurs différents : le MaltParser (Nivre *et al.*, 2006), le MSTParser (McDonald *et al.*, 2005b) [3], et FrMG (Villemonte de La Clergerie, 2005). Les deux premiers sont des parseurs statistiques entraînés sur le French Treebank, le dernier est un parseur symbolique fondé sur une extension du formalisme des grammaires d'arbres adjoints (Joshi *et al.*, 1975; Joshi et Schabes, 1997). Nous comparons les sorties de FrMG préalablement transformées pour respecter le schéma d'annotation du FTB. En effet, les sorties natives de FrMG présentent un schéma d'annotation beaucoup plus profond que celui du FTB.

Corpus Le corpus qui contient les cas de contrôle a été constitué sur le French Treebank et le SequoiaBank en utilisant des phrases issues du corpus d'entraînement. Or, ce corpus a servi à générer les modèles des deux parseurs statistiques. De fait, nous les avons réentraînés avec les mêmes paramètres que Candito *et al.* (2010), en enlevant les phrases concernées du corpus d'entraînement. Par ailleurs, le corpus d'entraînement utilise des parties du discours prédites par un tagger, grâce à une méthode de rééchantillonnage (jackknifing [4]). Pour des questions de commodité, nous avons utilisé les prédictions de Morfette (Chrupala *et al.*, 2008).

Métriques utilisées Nous utilisons les métriques standards d'évaluation d'analyses syntaxiques en dépendances, à savoir le *Labeled Attachment Score* (LAS) qui correspond au pourcentage de dépendances correctes, étiquette incluse et le *Unlabeled Attachment Score* (UAS), qui correspond au pourcentage de dépendances correctes, sans tenir compte des étiquettes sur les arcs. Comme il est d'usage, les évaluations sont faites sans tenir compte des ponctuations. De plus, nous donnons les métriques de référence que sont le rappel, la précision et le F_1-score pour évaluer notre système. On a alors une bonne représentation de ce que le système est capable d'ajouter comme dépendances et on peut aussi mesurer sa capacité à ne pas surgénérer (ajouter plus de dépendances qu'il ne faudrait).

3. L'architecture utilisée pour le parsing de nos corpus est l'architecture du projet Bonsai (Candito *et al.*, 2010), qui a adapté les deux parseurs statistiques au Français.

4. L'objectif du jackknifing est d'obtenir un treebank avec des parties du discours prédites par un tagger qui n'a pas été entraîné sur les données qu'il étiquette. En d'autres termes, on souhaite que le treebank possède autant d'erreurs que ce que prédirait le tagger sur un corpus arboré non modifié. On fait donc de la validation croisée dix fois, i.e. le corpus d'entraînement est divisé en dix parties, on entraîne le tagger sur neuf parties et on étiquette la dixième.

4.3 Mise au point des règles

Comme nous le mentionnions ci-dessus, les règles ont été mises au point sur les corpus de développement afin de ne pas biaiser leur application sur les corpus de test. En effet, il est important de les appliquer sur des phrases inconnues.

Les règles ont été écrites en utilisant au maximum le mécanisme de propagation par contraintes, afin d'en tester la robustesse et la généricité. Nous avons regardé divers exemples dans le corpus de développement et essayé d'en retirer les caractéristiques générales. A chaque fois qu'une règle était créée, nous l'appliquions à notre corpus de développement pour voir ce qu'elle couvrait. Nous regardions alors les divergences et mettions au point une nouvelle règle ou modifions celles existantes. Le tableau 3 récapitule le nombre de règles sur chaque corpus.

CORPUS	NOMBRE DE RÈGLES
Contrôle	3
Ellipse du sujet	1
Total	**4**

TABLE 3: Nombre de règles pour chaque corpus

Règles pour le contrôle sujet Nous avons une règle qui gère le contrôle sujet avec des verbes transitifs, par exemple : *Il veut venir* où *vouloir* est le verbe à contrôle et *venir* le verbe contrôlé.

Nous produisons une représentation graphique de notre règle à la figure 8. On cherche un verbe de catégorie **V** (et qui par ailleurs est connu pour être un verbe à contrôle) qui gouverne n'importe quel verbe à l'infinitif (de catégorie **VINF**) avec une étiquette de type *obj* ou *ats* ou *dep* ou *aux_caus*. Si une telle configuration est trouvée, alors nous attachons une contrainte **share_down** sujet sur l'arc.

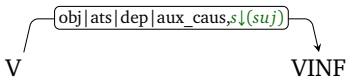


FIGURE 8: Règle pour le contrôle sujet avec des verbes transitifs

La deuxième règle gère le contrôle sujet des verbes intransitifs (*Il promet à Marie de venir*). La règle est illustrée à la figure 9. La configuration est sensiblement la même que pour 8, mais il faut prendre en compte l'ajout de la préposition. Si une telle configuration est trouvée, on ajoute un arc qui porte la contrainte **share_down** entre le verbe à contrôle et le verbe contrôlé.

Enfin, la troisième règle est particulière au verbe **venir** qui présente souvent une dépendance de type *mod* dans le FTB. Intégrer une telle dépendance à la première règle ferait baisser la précision du système.

Règle pour l'ellipse sujet La règle qui gère l'ellipse sujet est illustrée à la figure 10. Nous cherchons une configuration telle qu'un verbe possède un sujet et gouverne une conjonction

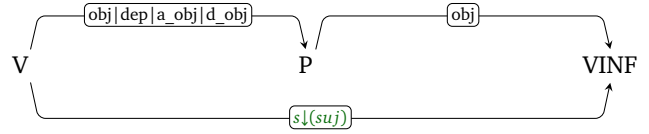


FIGURE 9: Règle pour le contrôle sujet avec des verbes intransitifs

de coordination qui elle-même gouverne un verbe (qui par ailleurs n'a pas de sujet). Si cette configuration existe, alors on crée un arc qui porte une contrainte **share_down** sujet entre les deux verbes. Les arcs gouvernés par le premier verbe seront partagés avec l'autre.

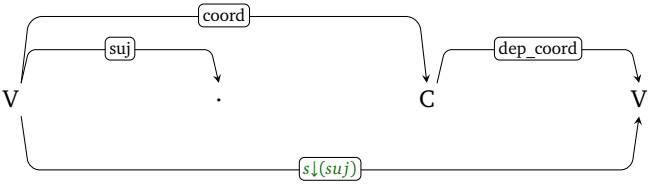


FIGURE 10: Règle pour l'ellipse sujet

L'idée qui se dessine avec la propagation de contraintes est que l'ordre d'application des règles est flexible donnant ainsi une plus grande liberté au système et à l'utilisateur qui écrit les règles. D'ailleurs, pour traiter les cas d'ellipse du sujet, il ne faudra que cette règle.

4.4 Résultats

Nous avons évalué deux choses différentes : la performance de nos règles et de notre système sur un corpus manuellement annoté et la généricité de nos règles sur des sorties d'analyseurs syntaxiques qui présentent, la plupart du temps, des erreurs par rapport à un corpus de référence.

Pour ce faire, nous avons appliqué nos règles sur notre corpus de référence sans les nouvelles annotations pour, ensuite, en évaluer l'impact. Les résultats sont reportés dans le tableau 4.

PHÉNOMÈNE	RAPPEL	PRÉCISION	F-SCORE
Contrôle	93,77	99,83	96,70
Ellipse du sujet	93,02	99,72	96,25

TABLE 4: Evaluation sur les corpus de test après applications des règles sur la référence (corpus gold)

Ensuite, pour mesurer la généricité de nos règles, il nous a paru intéressant de voir leur impact sur des sorties d'analyseurs syntaxiques. Nos deux corpus ont été parsés avec FrMG, mais aussi avec le MaltParser et le MSTParser. Nous avons évalué la performance des trois parseurs par rapport à notre corpus de référence. Les résultats sont donnés dans les tableaux 5 et 6.

	Dev		Test	
Parseur	*LAS*	*UAS*	*LAS*	*UAS*
FrMG	83,01	85,70	84,51	87,14
Malt	**86,50**	**89,41**	**87,55**	**90,05**
MST	84.34	87,48	85,82	88.65

Table 5: Contrôle : Evaluation des analyses de FrMG, de Malt et du MSTParser

	Dev		Test	
Parseur	*LAS*	*UAS*	*LAS*	*UAS*
FrMG	81,77	85,31	85,22	87,12
Malt	**85,31**	**87,35**	**86,05**	**88,07**
MST	83,54	86,77	83,45	85,89

Table 6: Ellipses : Evaluation des analyses de FrMG, de Malt et du MSTParser

Ensuite, nous avons appliqué notre système sur ces nouvelles analyses. Les résultats sont présentés dans les tableaux 7 et 8.

Parseur	LAS	UAS	Rappel	Précision	F-score
FrMG	84,24	86,77	85,81	99,64	92,21
Malt	87,07	89,50	83,91	99,71	91,13
MST	81,77	84,49	82,18	99,72	90,11
MST (Tagset réduit)	81,77	84,49	39,45	100,00	56,58

Table 7: Contrôle : Evaluation sur les sorties des trois parseurs après application des règles

Parseur	LAS	UAS	Rappel	Précision	F-score
FrMG	84,65	86,50	78,29	99,52	87,64
Malt	85,21	87,18	72,87	99,36	84,07
MST	82,56	84,97	68,21	99,39	80,91

Table 8: Ellipse : Evaluation sur les sorties des trois parseurs après application des règles

5 Discussion

Le tableau 4 donne les résultats lors de l'application des règles sur le corpus de référence. Les résultats montrent deux choses intéressantes :

1. La précision du système est excellente, ce qui signifie qu'il ne surgénère que très peu, n'ajoutant pas d'informations erronées au corpus.
2. Le rappel est certes moins bon, mais il reste tout de même haut : le système est capable d'ajouter un maximum de dépendances correctes.

En somme, sur le corpus de référence, les règles s'appliquent bien et ajoutent des dépendances qui nous permettent de nous approcher de la syntaxe profonde. De plus, on peut constater que pour des cas difficiles comme certaines coordinations elliptiques, le nombre d'erreurs n'est pas élevé.

Nous nous intéressons ensuite aux résultats sur les corpus analysés par les différents parseurs (tableaux 7 et 8) :

1. Nous obtenons de bons scores sur les analyses de FrMG, sans doute dû au fait que FrMG est un parseur produisant des analyses profondes, il est donc possible que les rattachements d'ellipses et de contrôle soient meilleurs.
2. Nous avons aussi de bons résultats sur les analyses de MaltParser pour les cas de contrôle. Cependant, les cas d'ellipses ne sont pas aussi concluants. En effet, Malt a des difficultés à retrouver les structures coordonnées, ce qui fait que les règles sont plus difficiles à appliquer ;
3. Concernant le MSTParser, dans une version préliminaire de ce papier, nous avions indiqué des résultats assez faibles concernant le rappel, indiquant que le parseur n'utilisait pas le même ensemble des parties du discours que les autres parseurs. En effet, suite à une erreur de paramétrage de notre part, le MSTParser a utilisé un ensemble plus restreint de parties du discours (V pour V, VINF, VPP, etc.). Les résultats présentés dans le tableau 7 corrigent ce problème. Néanmoins, par soucis d'exhausitivité nous rappelons à la ligne MSTParser (Tagset réduit) les anciens résultats obtenus.

L'utilisation du tagset réduit pour le MSTParser nous avait conduit à modifier nos règles dans ce sens, les rendant encore plus génériques. En effet, plus l'ensemble de parties du discours est réduit plus nous avons des chances d'appliquer nos règles sur un nombre plus important de structures, réduisant ainsi la précision du système. Nous avons donc évalué cet impact sur les différents corpus. Les résultats sont présentés dans le tableau 9.

Il est intéressant de constater que le fait de rendre les règles plus génériques entraîne une perte minime sur la référence, alors que nous avons un gain intéressant sur les analyses du MSTParser.

Au vue des résultats précédents, nous pouvons avancer que le système de propagation de contraintes fonctionne bien. Avec quelques contraintes, agissant sur des configurations simples, nous pouvons couvrir des phénomènes souvent difficiles à traiter, présentant des interactions. De plus, nous constatons que peu de règles suffisent à couvrir un phénomène et que les règles couvrant les cas particuliers sont évitées au maximum. En cela, le système reste générique. Par ailleurs, comme tout système à base de règles, il était important de voir à quel point les règles mises au point sur un corpus peuvent s'appliquer sur une version altérée de ce même corpus.

	LAS	UAS	RAPPEL	PRÉCISION	F-SCORE
FrMG	84,24	86,77	85,81	99,59	92,19
Malt	87,05	89.48	83,74	99,65	91,00
MST (Tagset réduit)	85,36	88,11	82,18	99,68	90,09
Référence	99,49	99,53	93,43	99,74	96,48

TABLE 9: CONTRÔLE : Evaluation après utilisation de règles plus génériques

Conclusion

A travers l'enrichissement d'un corpus arboré et de sortie d'analyseurs syntaxiques, nous avons montré qu'avec une approche générique de propagation par contraintes, il était possible d'avoir un système de réécriture de graphes plus facile à maintenir avec un nombre de règles restreint. Par ailleurs, grâce aux contraintes, il est possible d'exprimer des règles suffisamment génériques pour qu'elles puissent être utilisées sans modifications importantes sur des sorties prédites d'analyseurs syntaxiques.

Cela ouvre le champ à d'autres applications, parmi lesquelles nous pouvons citer la correction d'analyses syntaxiques afin d'améliorer les résultats d'un parseur. Dans la même mouvance, on peut tout autant essayer de réécrire des sorties d'analyseurs différents avec le même jeu de règles, pour ensuite comparer les graphes obtenus, afin de déduire le meilleur graphe à partir des différentes structures proposées. On pourrait améliorer les analyses syntaxiques au moyen de plusieurs parseurs et du système de réécriture qui guiderait les analyses.

Enfin, une autre piste à explorer, serait la transformation de schémas d'annotation dans le but d'évaluer les schémas entre eux. On sait que la comparaison de deux schémas d'annotations n'est pas chose aisée, ainsi utiliser un ensemble de règles, pour tenter de transformer un schéma en un autre permettrait une comparaison plus aisée des analyseurs syntaxiques entre eux.

Remerciements

Je souhaitais remercier Djamé Seddah et Éric de la Clergerie pour leurs conseils, leurs relectures attentives et leurs remarques lors de l'écriture de cet article.

Références

ABEILLÉ, A., CLÉMENT, L. et TOUSSENEL, F. (2003). Building a Treebank for French. *In Treebanks : Building and Using Parsed Corpora*, pages 165–188. Springer.

BONFANTE, G., GUILLAUME, B. et MOREY, M. (2011a). Modular graph rewriting to compute semantics. *In International Workshop on Computional Semantics 2011*.

BONFANTE, G., GUILLAUME, B., MOREY, M. et PERRIER, G. (2010). Réécriture de graphes de dépendances pour l'interface syntaxe-sémantique. *In TALN 2010*.

BONFANTE, G., GUILLAUME, B., MOREY, M. et PERRIER, G. (2011b). Enrichissement de structures en dépendances par réécriture de graphes. *In TALN 2011*.

BOUCHESÈCHE, L. (2009). Annotation semi-automatique des coordinations à ellipse sur corpus arboré. Mémoire de maîtrise, Univ. Paris Sorbonne 4.

CANDITO, M., NIVRE, J., DENIS, P. et HENESTROZA ANGUIANO, E. (2010). Benchmarking of Statistical Dependency Parsers for French. *In Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, Chine. Coling 2010.

CANDITO, M. et SEDDAH, D. (2012). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. *In TALN 2012 - 19e conférence sur le Traitement Automatique des Langues Naturelles*, Grenoble, France.

CHRUPALA, G., DINU, G. et van GENABITH, J. (2008). Learning Morphology with Morfette. *In Proceedings of LREC 2008*.

FRUEWIRTH, T. et ABDENNADHER, S. (2003). *Essentials of Constraint Programming*. Springer-Verlag New York, Inc.

GEISS, R., BATZ, G. V., GRUND, D., HACK, S. et SZALKOWSKI, A. (2006). GrGen : A fast SPO-Based graph rewriting tool. *In International Conference on Graph Transformation*.

JOSHI, A. K., LEVY, L. et TAKAHASHI, M. (1975). Tree Adjunct Grammars. *Journal of Computer and System Science 10*, 10(1).

JOSHI, A. K. et SCHABES, Y. (1997). Tree-adjoining grammars. *Handbook of formal languages*, 3:69–124.

LÖWE, M., EHRIG, H., HECKEL, R., RIBEIRO, L., WAGNER, A. et CORRADINI, A. (1993). Algebraic approches to graph transformations. *Theoritical Computer Science*.

MARCHAND, J., GUILLAUME, B. et PERRIER, G. (2010). Motifs de graphe pour le calcul de dépendances syntaxiques complètes. *In TALN 2010*.

MCDONALD, R., CRAMMER, K. et PEREIRA, F. (2005a). Online large-margin training of dependency parsers. *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

MCDONALD, R., PEREIRA, F., RIBAROV, K. et HAJIČ, J. (2005b). Non-projective dependency parsing using spanning tree algorithms. *In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

NIVRE, J., HALL, J. et NILSSON, J. (2006). MaltParser : A data-driven parser-generator for dependency parsing. *In Proc. of LREC-2006*.

RIBEYRE, C. (2012). Mise en place d'un systéme de réécriture de graphes appliqués á l'interface syntaxe-sémantique. Mémoire de master, Univ. Paris Diderot 7.

RIBEYRE, C., SEDDAH, D. et VILLEMONTE DE LA CLERGERIE, É. (2012). A Linguistically-motivated 2-stage Tree to Graph Transformation. *In* HAN, C.-H. et SATTA, G., éditeurs : *TAG+11 - The 11th International Workshop on Tree Adjoining Grammars and Related Formalisms - 2012*, Paris, France. INRIA.

SAGOT, B. (2010). The LEFFF, a freely available and large-coverage morphological and syntactic lexicon for french. *In Proceedings of LREC'10*, Valetta, Malta.

VILLEMONTE DE LA CLERGERIE, E. (2005). From metagrammars to factorized TAG/TIG parsers. *In Proceedings of IWPT'05 (poster)*, Vancouver, Canada.

A State of the Art of Word Sense Induction: A Way Towards Word Sense Disambiguation for Under-Resourced Languages

Mohammad Nasiruddin
Laboratoire d'Informatique de Grenoble-Groupe d'Étude pour la Traduction Automatique/Traitement Automatisé des Langues et de la Parole
Univ. Grenoble Alpes
mohammad.nasiruddin@imag.fr

ABSTRACT

Word Sense Disambiguation (WSD), the process of automatically identifying the meaning of a polysemous word in a sentence, is a fundamental task in Natural Language Processing (NLP). Progress in this approach to WSD opens up many promising developments in the field of NLP and its applications. Indeed, improvement over current performance levels could allow us to take a first step towards natural language understanding. Due to the lack of lexical resources it is sometimes difficult to perform WSD for under-resourced languages. This paper is an investigation on how to initiate research in WSD for under-resourced languages by applying Word Sense Induction (WSI) and suggests some interesting topics to focus on.

RÉSUMÉ

État de l'art de l'induction de sens: une voie vers la désambiguïsation lexicale pour les langues peu dotées

La désambiguïsation lexicale, le processus qui consiste à automatiquement identifier le ou les sens possible d'un mot polysémique dans un contexte donné, est une tâche fondamentale pour le Traitement Automatique des Langues (TAL). Le développement et l'amélioration des techniques de désambiguïsation lexicale ouvrent de nombreuses perspectives prometteuses pour le TAL. En effet, cela pourrait conduire à un changement paradigmatique en permettant de réaliser un premier pas vers la compréhension des langues naturelles. En raison du manque de ressources langagières, il est parfois difficile d'appliquer des techniques de désambiguïsation à des langues peu dotées. C'est pourquoi, nous nous intéressons ici, à enquêter sur comment avoir un début de recherche sur la désambiguïsation lexicale pour les langues peu dotées, en particulier en exploitant des techniques d'induction des sens de mots, ainsi que quelques suggestions de pistes intéressantes à explorer.

KEYWORDS: Word Sense Disambiguation, Word Sense Induction, under-resourced languages, lexical resources.
MOTS-CLÉS: désambiguïsation lexicale, induction de sens, langues peu dotées, ressources langagières.

1 Introduction

Word Sense Disambiguation (WSD) is a core and open research problem in Computational Linguistics and Natural Language Processing (NLP), which was recognized at the beginning of the scientific interest in Machine Translation (MT) and Artificial Intelligence (AI). On a variety of word types and ambiguities research has progressed steadily to the point where WSD systems achieve relatively high levels of accuracy.

The goal of a WSD is to computationally assign the correct sense of a word (i.e. meaning) in context (phrase, sentence, paragraph, text) from a predefined sense inventory, when the word has multiple meanings. It is a pervasive characteristic of natural language. The problem is that words often have more than one meaning, sometimes fairly similar and sometimes completely different. For example, the word *bank* has several senses and may refer to the edge of a river, a building, or a financial institution. The specific sense intended is determined by the textual context in which an instance of the ambiguous word appears. In "*The boy leapt from the bank into the cold water.*" the edge of a river is intended, whereas in "*The van pulled up outside the bank and three masked men got out.*" the building sense is meant, while in "*The bank sent me a letter.*" implies the financial institution sense.

Human readers have the capability to understand the meaning of a word from its context, but machines need to process textual information and transform it into data structures, which must then be analyzed in order to determine the underlying meaning. To perform WSD, a sense inventory must be available, which lists possible senses for the word of a text. A sense inventory is a lexical resource, which contains list of senses of a given word like the traditional dictionaries – knowledge resources. Manually annotated corpora with either word senses or information from knowledge sources is also an important resource for WSD.

Initially, WSD was mainly applied and developed on English texts, because of the broad availability and the prevalence of lexical resources compared to other languages. Due to the lack of lexical resources i.e. sense inventories (dictionaries, lexical databases, wordnets, etc.) and sense-tagged corpora it is difficult to start working on WSD for under-resourced languages (Bangla, Assamese, Oriya, Kannada, etc.). To account for under-resourced languages, one can easily adopt techniques aimed at the automatic discovery of word senses from text, a task called Word Sense Induction (WSI).

Languages with large amounts of data, or funding, or political interests can be interpreted as '*well-resourced*' languages, whereas, a lot of languages in the world do not enjoy this status, which is referred to in this article as '*under-resourced*' languages. This paper presents the state of the art of WSD and WSI in an under-resourced language perspective. In the following sections the paper is organized as follows: firstly, Sections 2 and 3 illustrate the main topics of interest in WSD and WSI respectively; then, Section 4 briefly describes WSI from an under-resourced language view; and finally, Section 5 concludes the article and discusses some perspectives for future work.

2 Word Sense Disambiguation

Depending on the degree of polysemy, there can be many different senses for a word and WSD algorithms aim at choosing the most appropriate sense combination among all possible senses for all words in a text unit (sentence, paragraph, etc.). It is essentially a task of classification for word of a text: word senses are the possible classes, the context provides evidence (features), and each occurrence of a word is assigned to one or more of its possible classes based on the evidence. There are many methods to perform WSD. In this section, various types of approaches and algorithms for WSD will be briefly presented.

The reader can refer to (Ide and Véronis, 1998) for works before 1998 and (Agirre and Edmonds, 2006), (Navigli, 2009) or (McCarthy, 2009) for a complete and current state of the art of WSD.

2.1 Approaches

WSD approaches can be categorized into supervised WSD and unsupervised WSD, and a further distinction can be made between knowledge-rich and knowledge-poor approaches (Navigli, 2009). Knowledge-rich methods involve the use of external knowledge sources whereas knowledge-poor methods do not. Based on machine learning techniques, researchers distinguish between supervised methods and unsupervised methods. There are three mainstream approaches to WSD, namely: Supervised WSD, Minimally-supervised WSD, and Unsupervised WSD. Figure 1 presents various WSD methods according to two axis: the quantity of annotated corpora required vertically and the amount of static knowledge horizontally.

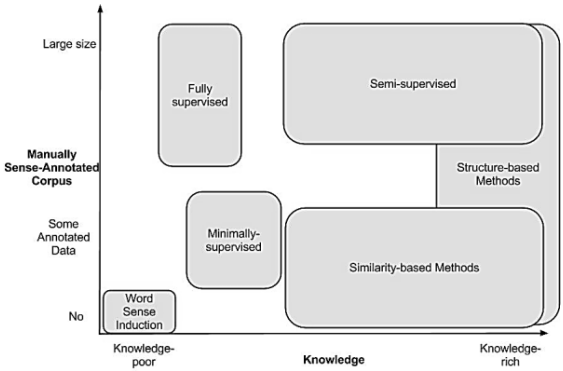


FIGURE 1 – Word Sense Disambiguation systems – Data versus Knowledge (Schwab, 2013 [personal notes]).

2.1.1 Supervised WSD

Supervised WSD uses supervised machine learning techniques. These approaches use a set of manually labeled training examples (i.e., sets of examples encoded as vectors whose elements represent features) to train a classifier for each target word. Support Vector Machines (SVMs), and memory-based learning have been shown to be the most successful approaches (Hoste *et al.*, 2002; Decadt *et al.*, 2004; Mihalcea *et al.*, 2004; Grozea, 2004; Chan *et al.*, 2007; Zhong and Ng, 2010), to date, probably because they can cope with the high-dimensionality of the feature space.

Supervised algorithms are progressively losing ground to the other methods. Moreover, they cannot easily be adapted to other languages without retraining (requires annotated data from that particular language). Furthermore, reusing models from one language for another leads at best to a poor classification performance (Khapra *et al.*, 2009).

2.1.2 Minimally-supervised WSD

Minimally supervised methods use a sense inventory, a few sense-annotated example instances, and raw corpora. From the sense-annotated examples, the system induces the senses or categories of senses for the non-annotated data, and then it functions exactly as an unsupervised clustering approach. The most prominent Minimally supervised method (Yarowsky, 1995), however, to our knowledge, has not been evaluated on SemEval WSD tasks.

2.1.2.1 Knowledge-based WSD

Knowledge-based WSD algorithms are similar to Minimally-supervised WSD approaches. The objective of Knowledge-based methods is to exploit static knowledge resources, such as dictionaries, thesauri, glossaries, ontologies, collocation etc., to infer the senses of words in context. Degree (Navigli and Lapata, 2010; Ponzetto and Navigli, 2010) or Personalized PageRank (Agirre and Soroa, 2009) are among the latest knowledge-based systems in the literature that exploits WordNet (Fellbaum, 1998) or other resources like BabelNet (Navigli and Ponzetto, 2010) to build a semantic graph and use the structural properties of the graph. In order to choose the appropriate senses of words knowledge-based systems use the structural properties of the graph in context either locally to the input sentence or globally.

2.1.3 Unsupervised WSD

Unsupervised learning is the greatest challenge for WSD researchers. Unsupervised WSD approaches are composed of Word Sense Induction or discrimination techniques aimed at discovering senses automatically based on unlabeled corpora and then applying them for WSD. By opposition to Supervised WSD, these approaches use machine learning techniques on non-sense-tagged corpora with no *a priori* knowledge about the task at all (see Section 4).

2.2 Evaluation

The evaluation of previous WSD algorithms is expressed in terms of the number of "correctly" disambiguated words as evaluated through a *Gold Standard (GS)*. Since 1998, there have been several follow-up evaluation campaigns (Senseval-1 (Kilgarriff, 1998), Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Mihalcea and Edmonds, 2004), SemEval-2007 (Navigli et al., 2007), SemEval-2010 (Agirre et al., 2010), SemEval-2012 (Manandhar and Yuret, 2012), and recently SemEval-2013 (Navigli et al., 2013)) with various disambiguation and semantic analysis tasks, which have been very successful and beneficial to the progress of the field.

In the evaluation task, a reference corpus is given with the lemmatized and part of speech (PoS) tagged instances (i.e. words), which will have to be disambiguated. The results are matched with the *GS* through *Precision (P)*, *Recall (R)*, and F_1 *score*, which are the standard-measures for evaluating WSD algorithms (Navigli, 2009). The evaluation tools provided calculate:

- P, the number of correct answers provided over the total number of answers provided;
- R, the number of correct answers provided over the total number of expected answers;
- F_1 *measure*, the harmonic mean between the two: $(2.P.R)/(P+R)$.

When all words are annotated by a WSD algorithm, then $P=R=F_1$.

3 Word Sense Induction

Word sense induction (WSI) is the task of automatically identifying the senses of words in texts, without the need of handcrafted resources or manually annotated data. It is an unsupervised WSD technique use machine learning methods on raw corpora without relying on any external resources such as dictionaries or sense-tagged data. During the learning phase, algorithms induce words senses from raw text by clustering word occurrences following the *Distributional Hypothesis* (Harris, 1954; Curran, 2004). This hypothesis was

popularized with the phrase "*a word is characterized by the company it keeps*" (Firth, 1957). Two words are considered semantically close if they co-occur with the same neighboring words. As a result, shifting the focus away from how to select the most suitable senses from an inventory towards how to automatically discover senses from a text. By applying WSI it is possible to mitigate the *Knowledge Acquisition Bottleneck* (Wagner, 2008) problem. The single common thread to WSI methods is the reliance on clustering algorithms used on the words in the unannotated corpus. Although the role of WSI, in a disambiguation context is to build a sense inventory that can be used subsequently for WSD, therefore WSI can be considered as part of WSD. Of course, WSI can have many more applications than building sense inventories and thus WSD.

3.1 Approaches

WSI algorithms extract the different senses of word following two approaches – locally and globally. Local algorithms discover senses of a word per-word basis i.e. by clustering its instances in contexts according to their semantic similarity, whereas global algorithms discovers senses in a global manner i.e. by comparing and determining them from the senses of other words in a full-blown word space model (Apidianaki and Van de Cruys, 2011). Based on the type of clustering algorithms used, will be reviewed various WSI proposed in the literature in the following subsections.

3.1.1 Clustering Approaches

Returning to the idea of (Harris, 1954; Curran, 2004) that word meaning can be derived from context, (Pantel and Lin, 2002) discover word senses from text. The underlying hypothesis of this approach is that words are semantically similar if they appear in similar documents, within similar context windows, or in similar syntactic contexts (Van de Cruys, 2010). *Lin's algorithm* (Lin, 1998) is a prototypical example of word clustering, which is based on *syntactic dependency statistics* between words that occur in a corpus to produce sets for each discovered sense of a target word (Van de Cruys and Apidianaki, 2011). By using a similarity function, the following clustering algorithms are applied to a test set of word feature vectors (Pantel and Lin, 2002): *K-means*, *Bisecting K-means* (Steinbatch *et al.*, 2000), *Average-link*, *Buckshot*, and *UNICON* (Lin and Pantel, 2001). *Clustering By Committee (CBC)* (Pantel and Lin, 2002) also uses syntactic contexts intended for the task of sense induction, but exploits a similarity matrix to encode the similarities between words. It relies on the notion of committees to output the different senses of the word of interest. However, These approaches are hard to apply on a large scale for many domains and languages.

3.1.2 Extended-clustering Approaches

Considering the observation that words tend to manifest one sense per collocation (Yarowsky, 1995), (Bordag, 2006) uses *word triplets* instead of word pairs. A well-known approach to extended-clustering is the *Context-group Discrimination* algorithm (Schütze, 1998) based on large matrix computation methods. Another approach, presented by (Pinto *et al.*, 2007), attempts to improve the usability of small, narrow-domain corpora through self-term expansion. (Brody and Lapata, 2009) shows that the task of word sense induction can also be framed in a *Bayesian* context by considering contexts of ambiguous words to be samples from a multinomial distribution. There are other extended-clustering approaches, that include the *bi-gram* clustering technique proposed by (Schütze, 1998), the clustering technique using

phrasal *co-occurrences* presented by (Dorow and Widdows, 2003), the technique for word clustering using a *context window* presented by (Ferret, 2004) and the method applying the *Information Bottleneck* algorithm for sense induction proposed by (Niu *et al.*, 2007). These additional clustering techniques can be broadly categorized as either improving feature selection and enriching features or introducing more effective and efficient clustering algorithms.

3.1.3 Graph-based Approaches

The main hypothesis of co-occurrence graphs is assuming that the semantic of a word is represented by means of co-occurrence graph, whose vertices are co-occurrences and edges are co-occurrence relations. These approaches are related to word clustering methods, where co-occurrences between words can be obtained on the basis of *grammatical* (Widdows and Dorow, 2002) or *collocational relations* (Véronis, 2004). (Klapaftis and Manandhar, 2007) propose the idea of the *Hypergraph* model for such WSI approaches. HyperLex (Véronis, 2004) is a successful graph based algorithm, based on the identification of hubs in co-occurrence graphs that have to cope with the need to tune a large number of parameters (Agirre *et al.*, 2006b). To deal with this issue several graph-based algorithms have been proposed, which are based on simple graph patterns, namely *Curvature Clustering* (Dorow *et al.*, 2005), *Squares, Triangles and Diamonds (SquaT++)* (Navigli, 2010), and *Balanced Maximum Spanning Tree Clustering (B-MST)* (Di Marco *et al.*, 2011). The patterns aim at identifying word meanings using the local structural properties of the co-occurrence graph. A randomized algorithm which partitions the graph vertices by iteratively transferring the mainstream message (i.e. word sense) to neighboring vertices proposed by (Biemann, 2006) is *Chinese Whispers*. By applying co-occurrence graph approaches, (Agirre *et al.*, 2006a; Agirre and Soroa, 2007; Korkontzelos and Manandhar, 2010) have been able to achieve state of the art performance in standard evaluation tasks. (Jurgens, 2011) reinterpret the challenge of identifying sense specific information in a co-occurrence graph as one of community detection, where a community is defined as a group of connected nodes that are more interconnected than to the rest of the graph (Fortunato, 2010). Recently, (Hope and Keller, 2013) introduced a linear time graph-based soft clustering algorithm for WSI named *MaxMax*, which obtains comparable results with those of systems based on existing, state of the art methods.

	Basic Word Co-occurrence	**Additional Features**
Classical Clustering Algorithms	Classical Clustering	Triplet Clustering Self-term Expansion Context Clustering Translation Features
Novel Algorithms for WSI	Clustering by Committee (CBC) Information Bottleneck	Hypergraph Collocation Graph Bayesian Model

TABLE 1 – Overview of Techniques for Word Sense Induction (Denkowski, 2009).

Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997; Landauer *et al.*, 1998) is currently a very popular approach to WSI that operates on word spaces (Van de Cruys and Apidianaki, 2011). *LAS* aims at finding and extract latent dimensions of meaning using *NMF (Non-negative Matrix Factorization)*, *PCA (Principal Component Analysis)* or *SVD (Singular Value Decomposition)*. The extracted latent dimensions are then used to distinguish between

different senses of a target word that are in turn used to disambiguate each given instance of that word.

3.1.4 Translation-oriented Approaches

WSI approaches described above cover only monolingual data; in the context of Machine Translation (MT), recent work has been done to incorporate bilingual data into the sense induction task. Translation-oriented WSI approaches involve augmenting the source language context with target language equivalents. (Apidianaki, 2008) describes this process by using a bilingual corpus that has been word aligned by type and token to construct two bilingual dictionaries, where each word type is associated with its translation equivalent. The lexicon is filtered such a way that words and their translation equivalents have matching PoS tags and words appear in the translation lexicons for both directions.

3.2 Evaluation

The evaluation of WSI approaches is one of the key challenges for researchers. As the sense clusters derived by these algorithms may not match the actual senses defined in lexical resources like dictionaries, lexical databases, wordnets, etc., the evaluation of these algorithms needs to be carried out manually, by asking language experts to corroborate the results. However, it is hard to evaluate the results of WSI, as the resulting clustered senses can vary from algorithm to algorithm or even for various parameter values for a single algorithm, as even determining the number of clusters a difficult matter. From the very beginning of WSI, depending on different approaches researchers have developed various evaluation methodologies that can be separated into three main categories.

3.2.1 Supervised Evaluation

In this evaluation method, the target word corpus is divided into two parts – a testing and a training part. Firstly, the training part is used to map the automatically induced clusters to *Gold Standard (GS)* senses. In the next step, the test corpus is used to evaluate WSI approaches in a WSD (Agirre and Soroa, 2007) setting. Finally, the usual *Precision (P)* and *Recall (R)* are used to determine the quality of the resulting WSD.

3.2.2 Unsupervised Evaluation

In this evaluation setting, the induced senses are evaluated as clusters of examples, and compared to sets of examples that have been tagged with *sense labels from a Gold Standard*. The *V-measure* (Rosenberg and Hirschberg, 2007) is used to determine the quality of clusters by combining metrics such as the *Paired F-Score* (Manandhar *et al.*, 2010), the *RandIndex* (Rand, 1971; Navigli, 2010) and others. They measure both the coverage and the homogeneity of a clustering output as opposed to the traditional clustering measure of *F-Score* (Zhao *et al.*, 2005) that is most commonly used to assess the performance of WSI systems.

3.2.3 Evaluation as an Application

Recently, (Navigli and Crisafulli, 2010; Di Marco and Navigli, 2013) proposed to evaluate WSI approaches as part of a specific application, where WSI techniques have been shown to consistently surpass symbolic state of the art systems (See section 4.3). The evaluation of WSI and WSD systems was performed in the context of *Web search result clustering*.

3.3 SemEval WSI Evaluation Tasks

This section briefly describes the different SemEval workshops (from 2007 to 2013) with WSI evaluation tasks that focus on the evaluation of semantic analysis systems.

3.3.1 SemEval-2007 Task 2

The goal of SemEval-2007 Task 2 (Agirre and Soroa, 2007) is to allow for the comparison across sense-induction and discrimination systems, and also to compare these systems to other supervised and knowledge-based systems. This task evaluates WSI systems on 33 nouns and 65 verbs (lexical sample), where the corpus consists of texts of the *Wall Street Journal (WSJ)* corpus, and is hand-tagged with *OntoNotes* senses (Hovy *et al.*, 2006). For each tagged-word, the task consists of first identifying the senses of target words (e.g. as clusters of target word instances, co-occurring words, etc.), and secondly tagging the instances of the target word using the automatically induced clusters. This double evaluation methodology (i.e. supervised evaluation and unsupervised evaluation) has been attempted by (Agirre *et al.*, 2006a).

3.3.2 SemEval-2010 Task 14

SemEval-2010 Task 14 is a continuation of the WSI SemEval-2007 Task 2 with some significant changes to the evaluation setting. The main difference in this task compared to the SemEval-2007 WSI task, is that the training and testing data are treated separately, which allows for a more realistic evaluation of the clustering models. Readers may refer to (Klapaftis and Manandhar, 2013) for a detailed analysis of the SemEval-2010 WSI task evaluation result and new evaluation settings.

3.3.3 SemEval-2013 Task 11

For the evaluation in SemEval-2013: Task 11, WSI and WSD systems are applied to web search result clustering, where the test data consists of 100 topics (all nouns), each with a list of 64 top-ranking documents. The topics were selected from the list of ambiguous *Wikipedia* entries (i.e., those with "*disambiguation*" in the title) among queries of lengths ranging between 1 and 4 words. The 64 snippets associated with each topic were collected from the results provided by the *Google* search engine. Three annotators tagged each snippet with the most appropriate meaning from *Wikipedia*, with adjudication in the case of disagreement. For a detailed description of the SemEval-2013: Task 11 evaluations please refer to (Di Marco and Navigli, 2013; Navigli and Vannella, 2013).

4 WSI for Under-Resourced Languages

Given the above-mentioned difficulties, WSI is an attractive alternative to WSD, especially for under-resourced languages.

The Figure 2 shows an overview of the state of freely available resources for a certain number of representative languages and the aim of the diagram is to give the reader an idea of the current situation. Making a highly precise diagram would be prohibitively complex and furthermore, the position of the various languages must be interpreted relatively to each other rather than in an absolute manner (except for English that we placed in the top right-hand corner as a reference).

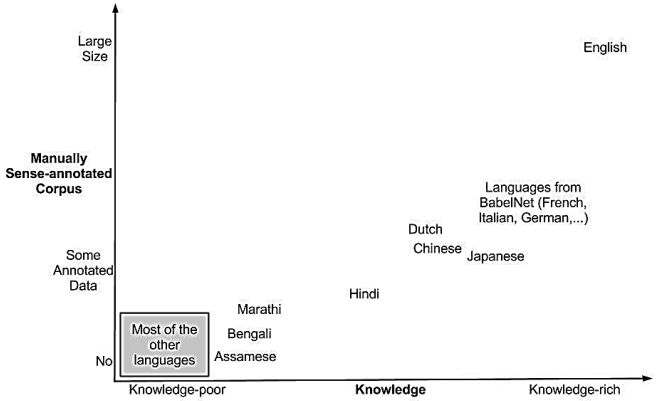


FIGURE 2 – Computationally language richness – Data versus Knowledge (Schwab, 2013 [personal notes]).

As described previously, two kinds of resources are difficult/expensive to build: annotated corpora and lexical databases (static versus dynamic knowledge). While lexical databases can be used directly in many applications or by humans, sense-annotated corpora have less applications and are much more expensive to build. An automatic procedure can extract the senses that are objectively present in a particular corpus and allows for the sense inventory to be straightforwardly adapted to a new domain. By applying WSI, it is practical to disambiguate particular word instances using the automatically extracted sense inventory. Words and contexts are mapped to a limited number of topic dimensions (depending on the topic of the words and contexts) in a latent semantic word space. A particular sense is associated with a particular topic and different senses can be discriminated through their association with particular topic dimensions. (Van de Cruys and Apidianaki, 2011) describe the induction step and disambiguation step as being based on the same principle.

Currently, *Wikipedia* contains 285 languages, anyone can generate corpus from a *Wikipedia dump*, or blogs, forums, newspaper articles in any language. As WSI is a kind of clustering problem, the evaluation of the clusters is normally difficult, however if the evaluation process is followed, it becomes rather straightforward. Though semantic evaluation campaigns are based on some dominant languages, progress for under-resourced languages is still ongoing. In this regard, Crowdsourcing (Sabou et al., 2012), especially *Games With A Purpose* (von Ahn, 2006) are considered as an attractive alternative for collecting annotated data (Wang et al., 2010), which can be subsequently used as a *GS* for evaluating the systems.

5 Conclusion and Discussions

The state of the art of Word Sense Disambiguation followed by Word Sense Induction techniques for under-resourced languages are provided in this article, and also tries to provide a basic idea of the essentials of the field. Here, the authors show a way to work on WSD by using WSI approaches in an unsupervised way, where very few resources (like corpora) are available. Basically, the performance of WSD systems depends heavily on which sense inventory is chosen, here WSI overcomes this issue by allowing unrestrained sets of senses. Besides, its evaluation is particularly hard because there is no easy way of comparing and ranking different representations of senses.

Acknowledgements

The work presented in this paper was conducted in the context of the VideoSense project, funded by the French National Research Agency (ANR) under its CONTINT 2009 program (grant ANR-09-CORD-026).

References

AGIRRE, E., LOPEZ DE LACALLE, O., AND SOROA, A. (2009). Knowledge-Based WSD on Specific Domains: Performing Better than Generic Supervised WSD. *In Proceedings of the 21st Int'l Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1501– 1506. California.

AGIRRE, E., MARTINEZ, D., LOPEZ DE LACALLE, O., AND SOROA, A. (2006a). Two Graph-Based Algorithms for State of the Art WSD. *In Proceedings of the Conference on EMNLP*, pp. 585-593.

AGIRRE, E., MARTÍNEZ, D., LÓPEZ DE LACALLE, O. AND SOROA, A. (2006b). Evaluating and Optimizing the Parameters of an Unsupervised Graph-Based WSD Algorithm. *In Proceedings of TextGraphs: the 2nd Workshop on Graph Based Methods for NLP*, pp. 89–96. New York, USA.

AGIRRE, E. AND SOROA, A. (2007). SemEval-2007 Task 2: Evaluating Word Sense Induction and Discrimination Systems. *In Proceedings of the 4th Int'l Workshop on SemEval-2007*, pp. 7–12.

AGIRRE, E. AND SOROA, A. (2009). Personalizing PageRank for Word Sense Disambiguation. *In Proceedings of the 12th Conference of the EACL 2009*, pp. 33–41. Athens, Greece.

AGIRRE, E., LOPEZ DE LACALLE, O., FELLBAUM, C., HSIEH, S.K., TESCONI, M., MONACHINI, M., VOSSEN, P. AND SEGERS, R. (2010). SemEval-2010 Task 17: All-Words Word Sense Disambiguation on a Specific Domain. *In Proceedings of the 5th Int'l Workshop on Semantic Evaluation*, pp. 75–80.

APIDIANAKI, M. (2008). Translation-Oriented Word Sense Induction Based on Parallel Corpora. *In Proceedings of the 6th Int'l Conference on Language Resources and Evaluation*. Morocco.

APIDIANAKI, M. AND VAN DE CRUYS, T. (2011). A Quantitative Evaluation of Global Word Sense Induction. *In Proceedings of the 12th Int'l Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2011)*, pp. 253–264. Tokyo, Japan.

BALDWIN, T., KIM, S., BOND, F., FUJITA, S., MARTINEZ, D. AND TANAKA, T. (2010). A Reexamination of MRD-Based Word Sense Disambiguation. *In ACM Transactions on Asian Language Information Processing (TALIP) 9*, pp. 4:1–4:21. ACM.

BIEMANN, C. (2006). Chinese Whispers – An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems. *In Proceedings of the TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pp. 73–80, USA.

BORDAG, S. (2006). Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. *In Proceedings of the 11th Conference of the EACL 2006*, pp. 137–144. Trento, Italy.

BRODY, S. AND LAPATA, M. (2009). Bayesian Word Sense Induction. *In Proceedings of the 12th Conference of the EACL 2009*, pp. 103–111. Athens, Greece.

CHAN, Y.S., NG, H.T. AND ZHONG, Z. (2007). NUS-PT: Exploiting Parallel Texts for Word Sense Disambiguation in the English All-Words Tasks. *In Proceedings of the 4th Int'l Workshop on Semantic Evaluations (SemEval-2007)*, pp. 253–256. Prague, Czech Republic.

CURRAN, J. R. (2004). PhD Thesis: From distributional to semantic similarity. *University of Edinburgh*. Edinburg, UK.

DECADT, B., HOSTE, V., DAELEMANS, W. AND VAN DEN BOSCH, A. (2004). GAMBL, Genetic Algorithm Optimization of Memory-Based WSD. *In Proceedings of the 3rd Int'l Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 108–112. Barcelona, Spain.

DENKOWSKI, M. (2009). A Survey of Techniques for Unsupervised Word Sense Induction. *In Language & Statistics II Literature Review*.

DI MARCO, A. AND NAVIGLI, R. (2011). Clustering Web Search Results With Maximum Spanning Trees. *In Proceedings of the 12th Int'l Conference of the Italian Association for AI*, pp.201–212.

DI MARCO, A. AND NAVIGLI, R. (2013). Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction. *In Computational Linguistics 39(4)*, pp. 201–212. MIT.

DOROW, B. AND WIDDOWS, D. (2003). Discovering Corpus-Specific Word Senses. *In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, pp. 79–82. Budapest, Hungary.

DOROW, B., WIDDOWS, D., LING, K., ECKMANN, J., SERGI, D. AND MOSES, E. (2005). Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination. *In Proceedings of the MEANING-2005 Workshop*.

EDMONDS, P. AND COTTON, S. (2001). SENSEVAL-2: Overview. *In Proceedings of the 2nd Int'l Workshop on Evaluating Word Sense Disambiguation Systems, pp. 1-5*. France.

FELLBAUM, C. (ED.) (1998). WordNet: An Electronic Database. *MIT Press*. Cambridge, M.A, USA.

FERRET, O. (2004). Discovering Word Senses from a Network of Lexical Cooccurrences. *In Proceedings of the 20th Int'l Conference on Computational Linguistics*, pp. 1326–1332.

FIRTH, J.R. (1957). A synopsis of linguistic theory 1930-1955. *In Studies in Linguistic Analysis (Oxford: Philological Society)*, pp. 1–32. Reprinted in Palmer, F.R., (ed.) (1968). Selected Papers of J.R. Firth 1952-1959. London: Longman.

FORTUNATO, S. (2010). Community Detection in Graphs. *In Physics Reports 486*, pp. 75–174.

GROZEA, C. (2004). Finding Optimal Parameter Settings for High Performance Word Sense Disambiguation. *In Proceedings of the 3rd Int'l Workshop on the Senseval-3*, pp. 125–128.

HARRIS, Z. (1954). Distributional Structure. *In Papers in Structural and Transformational Linguistics*, pp. 775–794.

HOPE, D. AND KELLER, B. (2013). MaxMax: A Graph-Based Soft Clustering Algorithm Applied to Word Sense Induction. *In Proceedings of the Int'l Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2013)*, pp. 368–381. Samos, Greece.

HOSTE, V., HENDRICKX, I., DAELEMANS, W. AND VAN DEN BOSCH, A. (2002). Parameter Optimization for Machine-Learning of Word Sense Disambiguation. *In Natural Language Engineering 8(04)*, pp. 311–325. Cambridge University Press.

HOVY, E., MARCUS, M., PALMER, M., RAMSHAW, L. AND WEISCHEDEL, R. (2006). OntoNotes: The 90% Solution. *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 57–60. USA.

IDE, N. AND VÉRONIS, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *In Computational Linguistics 24(1)*, pp. 02–40. Cambridge, MA, USA.

IDE, N., ERJAVEC, T. AND TUFIS, D. (2002). Sense Discrimination with Parallel Corpora. *In Proceedings of ACL-02 Workshop on Word Sense Disambiguation*, pp. 61–66. USA.

JABBARI, S., HEPPLE, M. AND GUTHRIE, L. (2010). Evaluation Metrics for the Lexical Substitution Task. *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 289–292. California, USA.

JIN, P., WU, Y. AND YU, S. (2007). SemEval-2007 Task 05: Multilingual Chinese-English Lexical Sample. *In Proceedings of the 4th Int'l Workshop on Semantic Evaluations (SemEval-2007)*, pp. 19–23. Prague, Czech Republic.

JURGENS, D. (2011). Word Sense Induction by Community Detection. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics Human Language Technologies (ACL HLT 2011)*, pp. 24–28. Portland, Oregon, USA.

KHAPRA, M.M., SHAH, S., KEDIA, P., AND BHATTACHARYYA, P. (2009). Projecting Parameters for Multilingual Word Sense Disambiguation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pp. 459—467. Singapore.

KHAPRA, M.M., KULKARNI, A., SOHONEY, S. AND BHATTACHARYYA, P. (2010). All Words Domain Adapted WSD: Finding a Middle Ground Between Supervision and Supervision. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1532–1541.

KILGARRIFF, A. (1998). SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. *In Proceeding of the 1st Int'l Conference on Language Resources and Evaluation (LREC 1998)*, pp. 581—588. Granada, Spain.

KLAPAFTIS, I.P. AND MANANDHAR, S. (2007). UoY: A Hypergraph Model for Word Sense Induction and Disambiguation. *In Proceedings of the 4th Int'l Workshop on Semantic Evaluations (SemEval-2007)*, pp. 414–417. Prague, Czech Republic.

KLAPAFTIS, I.P. AND MANANDHAR, S. (2013). Evaluating Word Sense Induction and Disambiguation Methods. *In Language Resources and Evaluation*, pp. 1–27. Springer.

KOELING, R., MCCARTHY, D. AND CARROLL, J. (2005). Domain-Specific Sense Distributions and Predominant Sense Acquisition. *In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in NLP*, pp. 419–426. B.C., Canada.

KORKONTZELOS, I. AND MANANDHAR, S. (2010). UoY: Graphs of Unambiguous Vertices for Word Sense Induction and Disambiguation. *In Proceedings of the 5th Int'l Workshop on Semantic Evaluation (SemEval-2010)*, pp. 355–358, Uppsala, Sweden.

LANDAUER, T.K. AND DUMAIS, S.T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *In Psychology Review (104)*, pp. 211–240.

LANDAUER, T., FOLTZ, P. AND LAHAM, D. (1998). An Introduction to Latent Semantic Analysis. *In Discourse Processes*, pp. 25: 284–295.

LIN, D. (1998). Automatic Retrieval and Clustering of Similar Words. *In Proceedings of the 17th Int'l Conference on Computational Linguistics,* pp. 768–774. Quebec, Canada.

LIN, D. AND PANTEL, P. (2001). DIRT–Discovery of Inference Rules from Text. *In Proceedings of the 7th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pp. 323–328.

MANANDHAR, S., KLAPAFTIS, I.P., DLIGACH, D. AND PRADHAN, S.S. (2010). SemEval-2010 Task 14: Word Sense Induction & Disambiguation. *In Proceedings of the 5th Int'l Workshop on Semantic Evaluation (SemEval-2010)*, pp. 63–68. Uppsala, Sweden.

MANANDHAR, S. AND YURET, D. (2012). SemEval-2012: Semantic Evaluation Exercises. *In Proceedings of the SemEval-2012: Semantic Evaluation Exercises*. Montreal, Canada.

MCCARTHY, D. AND NAVIGLI, R. (2009). The English Lexical Substitution Task. *In Language Resources and Evaluation 43(2)*, pp. 139–159. Springer.

MIHALCEA, R. AND EDMONDS, P. (2004). Senseval-3: The Third Int'l Workshop on the Evaluation of Systems for the Semantic Analysis of Text. *In Proceedings of Senseval-3: The 3rd Int'l Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

MIHALCEA, R. AND FARUQUE, E. (2004). Senselearner: Minimally Supervised Word Sense Disambiguation for All Words in Open Text. *In Proceedings of ACL/SIGLEX*, pp. 155–158.

NAVIGLI, R. (2009). Word Sense Disambiguation: A Survey. *In ACM Computing Surveys (CSUR) 41(2)*, pp. 1–69. ACM.

NAVIGLI, R. AND CRISAFULLI, G. (2010). Inducing Word Senses to Improve Web Search Result Clustering. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-10)*, pp. 116–126. Boston, USA.

NAVIGLI R., JURGENS, D. AND VANNELLA, D. (2013). SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. *In Proceedings of the 7th Int'l Workshop on Semantic Evaluation, the 2nd Joint Conference on Lexical and Computational Semantics*, pp. 116–126. Atlanta, GA, USA.

NAVIGLI, R. AND LAPATA, M. (2010). An Experimental Study on Graph Connectivity for Unsupervised Word Sense Disambiguation. *In IEEE Transactions on Pattern Analysis and Machine Intelligence 32(4)*, pp. 678–692. IEEE.

NAVIGLI, R., LITKOWSKI, K.C. AND HARGRAVES, O. (2007). SemEval-2007 Task 07: Coarse-Grained English All-Words Task. *In Proceedings of 4th Int'l Workshop on SemEval-2007*, pp. 30–35.

NAVIGLI, R. AND PONZETTO, S.P. (2010). BabelNet: Building a Very Large Multilingual Semantic Network. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 216–225. Uppsala, Sweden.

NAVIGLI, R. AND VANNELLA, D. (2013). SemEval-2013 Task 11: Word Sense Induction & Disambiguation within an End-User Application. *In Proceedings of the 7th Int'l Workshop on Semantic Evaluations (SemEval-2013)*. Atlanta, GA, USA.

NIU, Z., JI, D. AND TAN, C. (2007). I2R: Three Systems for Word Sense Discrimination Chinese Word Sense Disambiguation and English Word Sense Disambiguation. *In Proceedings of the 4th Int'l Workshop on Semantic Evaluations,* pp. 177–182. Prague, Czech Republic.

PANTEL, P. AND LIN, D. (2002). Discovering Word Senses from Text. *In Proceedings of the 8th Int'l Conference on Knowledge Discovery and Data Mining*, pp. 613–619. Canada.

PINTO, D., ROSSO, P. AND JIMENEZ-SALAZAR, H. (2007). UPV-SI: Word Sense Induction Using Self-Term Expansion. *In Proceedings of 4th Int'l Workshop on Semantic Evaluations*, pp. 430–433.

PONZETTO, S.P. AND NAVIGLI, R. (2010). Knowledge-Rich Word Sense Disambiguation Rivaling Supervised System. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 1522–1531. Uppsala, Sweden.

RAND, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *In Journal of the American Statistical Association 66(336)*, pp. 846–850. Taylor & Francis.

ROSENBERG, A. AND HIRSCHBERG, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 410–420.

SABOU, M., BONTCHEVA, K. AND SCHARL, A. (2012). Crowdsourcing Research Opportunities: Lessons from Natural Language Processing. *In Proceedings of the 12th Int'l Conference on Knowledge Management and Knowledge Technologies*, pp. 17:1–17:8.

SCHÜTZE, H. (1998). Automatic Word Sense Discrimination. *In Computational Linguistics 24(1)*, pp. 97–124. MIT Press.

STEINBACH, M., KARYPIS, G., AND KUMAR, V. (2000). A Comparison of Document Clustering Techniques. *In Proceedings of the 6th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pp. 525–526. Boston, USA.

VAN DE CRUYS, T. AND APIDIANAKI, M. (2011). Latent Semantic Word Sense Induction and Disambiguation. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1476– 1485. Oregon, USA.

VAN DE CRUYS, T. (2010). PhD Thesis: Mining for Meaning – the Extraction of Lexico-Semantic Knowledge from Text. *University of Groningen*, pp. 12–18. The Netherlands.

VÉRONIS, J. (2004). Hyperlex: Lexical Cartography for Information Retrieval. *In Computer Speech and Language 18(3)*, pp. 223–252.

VON AHN, L. (2006). Games With A Purpose. *In Computer 6(39)*, pp. 92–94. IEEE Computer Society Press.

WAGNER, C. (2006). Breaking the knowledge acquisition bottleneck through conversational knowledge management. *In Information Resources Management Journal (IRMJ) 19(1)*, pp. 70–83. IGI Global.

WANG, A., HOANG, C.D.V. AND KAN, M.Y. (2004). Perspectives on Crowdsourcing Annotations for Natural Language Processing. *In Language Resources and Evaluation*, pp. 1–23. Springer.

WIDDOWS, D. AND DOROW, B. (2002). A Graph Model for Unsupervised Lexical Acquisition. *In Proceedings of the 19th Int'l Conference on Computational Linguistics*, pp. 1–7.

YAROWSKY, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995)*, pp. 189–196. Cambridge, Massachusetts, USA.

ZHAO, Y., KARYPIS, G., AND FAYYAD, U. (2005). Hierarchical Clustering Algorithms for Document Datasets. *In Data Mining and Knowledge Discovery, 10(2)*, pp. 141–168. The Netherlands.

ZHONG, Z. AND NG, H.T. (2010). It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 78–83. Uppsala, Sweden.

Génération de corpus en dialecte tunisien pour l'adaptation de modèles de langage

Rahma Boujelbane[1,2]
(1) ANLP_MIRACL, Sfax, Tunisie
(2) LIF, UMR7279, 13288, Marseille, France
Rahma.boujelbane@gmail.com

RÉSUMÉ

Ces derniers temps, vu la situation préoccupante du monde arabe, les dialectes arabes et notamment le dialecte tunisien est devenu de plus en plus utilisé dans les interviews, les journaux télévisés et les émissions de débats. Cependant, cette situation présente des conséquences négatives importantes pour le Traitement Automatique du Langage Naturel (TALN): depuis que les dialectes parlés ne sont pas officiellement écrits et n'ont pas d'orthographe standard, il est très coûteux d'obtenir des corpus adéquats à utiliser pour des outils de TALN. Par conséquent, il n'existe pas des corpus parallèles entre l'Arabe Standard Moderne(ASM) et le Dialecte Tunisien (DT). Dans ce travail, nous proposons une méthode pour la création d'un lexique bilingue ASM–DT et un processus pour la génération automatique de corpus dialectaux. Ces ressources vont servir à la construction d'un modèle de langage pour les journaux télévisés tunisiens, afin de l'intégrer dans un Système de Reconnaissance Automatique de Parole (SRAP).

ABSTRACT

Generation of tunisian dialect corpora for adapting language models.

Lately, given the serious situation in the Arab world, the Arab dialects such as Tunisian dialect became increasingly used and represented in the interviews, news and debate programs. However, this situation presents negative consequences for Natural Language Processing (NLP): Since dialects are not officially written and have no orthographic standard, it is very costly to obtain adequate corpora to train NLP tools. Therefore, it does not even exist parallel corpora between Standard Arabic (MSA) and Tunisian Dialect (TD). In this work, we propose a method for the creation of a bilingual lexicon MSA-TD and an automatic process for generating dialectal corpora. These resources will be used to build a language model for Tunisian news, in order to integrate it into an Automatic Speech Recognition (ASR).

MOTS-CLÉS : Dialecte Tunisien, lexique ASM-DT, TDT: Tunisian Dialect Translator.
KEYWORDS : Tunisian Dialect, MSA-TD lexicon, TDT: Tunisian Dialect Translator.

1 Introduction

L'utilisation de corpus constitue un problème crucial pour les langues disposant peu de ressources électroniques et peu informatisées comme de nombreux dialectes arabes. En effet, la construction de corpus est une étape capitale pour une bonne réalisation d'outils de traitement automatique de la langue tels que les systèmes de reconnaissance de parole qui nécessitent des données textuelles en grande quantité pour apprendre le vocabulaire d'une langue. Récemment en Tunisie, la révolution a touché non seulement le peuple

mais aussi les médias. Par conséquent, en un an tout le paysage médiatique a été bouleversé: les chaines, les émissions de débats et les journaux télévisés se sont multipliés. Ceci a donné naissance à un nouveau type de discours médiatique. En effet, la majorité des discours ne sont plus en ASM mais ils présentent une alternance entre le ASM et le dialecte. En effet, nous pourrons distinguer dans un même discours des mots en ASM, des mots en DT et des mots ASM «dialectalisés» tel qu'un mot avec une racine ASM et des affixes dialectales. Face à cette situation un SRAP conçu pour le ASM serait incapable de transcrire cette nouvelle langue. Pour cela, nous focalisons dans le présent travail à construire des ressources représentatives de ce mélange entre le ASM et le DT. Pour ce faire, nous proposons une approche basée sur deux étapes principales: La première consiste à construire une base lexicale, dans laquelle nous introduisons des règles de correspondance entre des structures en arabe standard et des structures dialectales. La deuxième étape se repose sur l'exploitation de cette base lexicale afin de générer des corpus dialectaux.

2 Travaux connexes

Les langues orales qui n'ont pas de forme écrite répandue peuvent être classées comme des langues peu dotées. De ce fait, plusieurs travaux ont tenté de pallier les problèmes liés à l'informatisation des langues peu dotées. **(Y. Scherrer, 2008),** dont le but d'informatiser le dialecte existant en Suisse, a développé un système de traduction allemand standard-suisse allemand. Le système développé traduit en se basant, sur un lexique bilingue, l'allemand standard vers n'importe quelle variété du continuum dialectal de la Suisse alémanique. Par ailleurs, les auteurs dans **(Shaalan et al., 2007**) se sont intéressés à l'informatisation du dialecte égyptien, l'une des variantes de l'arabe standard. Les auteurs ont proposé un système de traduction dialecte EGYptien EGY-ASM. A cette fin, ils ont essayé de construire un corpus parallèle EGY-ASM, ceci en se basant sur des règles de correspondance EGY- ASM. A part les dialectes, il existe plusieurs langues parmi la famille des langues peu dotées qui n'ont pas de relation avec une langue bien dotée comme le somalien et le khmer, etc. Ainsi, l'approche proposée pour constituer des corpus en somalien dans **(Nimaan et al., 2006)** se repose sur plusieurs scénarios: collecte de corpus à partir du Web, synthèse automatique de textes et traduction automatique français-somali. Et pour solliciter le manque de ressources en khmer **(Seng, 2010)** a choisi les sites de nouvelles en khmer au fort contenu rédactionnel pour collecter les corpus textuels. La revue de la littérature nous a montré qu'il n'y a pas assez de travaux qui ont traité l'arabe tunisien, la langue cible de ce travail. Le travail de **(Graja et al, 2011**) par exemple traite le dialecte tunisien pour la compréhension de parole. Cependant, ce travail utilise un domaine limité à savoir le transport ferroviaire où le vocabulaire est assez limité. Pour disposer des données, les chercheurs ne se sont basés que sur des transcriptions manuelles de conversations entre l'agent du guichet et les voyageurs. Or, un vocabulaire limité pose un problème si nous souhaitons modéliser un modèle de langage pour un système de reconnaissance des émissions de la télévision ayant un vocabulaire large et varié.

3 Ressources pour le dialecte tunisien

Le dialecte tunisien est une langue arabe rattachée à l'arabe maghrébin parlée par douze

millions de personnes vivant principalement en Tunisie. Bien que la langue officielle soit l'arabe littéral, il est généralement connu de ses locuteurs sous le nom de 'Darija' ou 'Tounsi' ce qui signifie tout simplement «tunisien», afin de le distinguer de l'arabe littéral (Baccouche, 1994). Dans les deux dernières années, ce dialecte est devenu la langue parlée dans la plupart des médias au lieu de l'arabe standard. Mais, cette forme dialectale a une forme sophistiquée: elle présente des formes mixtes ASM-DT et elle est en même temps un dialecte et une langue proche du ASM. Ainsi, étant donné l'écart faible entre cette forme dialectale et le ASM, les ressources disponibles pour le ASM peuvent être avantageusement utilisées pour créer des ressources dialectales.

3.1 L'Arabic TreeBank pour la création d'un lexique bilingue ASM-DT

Au début de cette étude et lors d'une convention avec le LDC (Linguistic Data Consortium), nous avons eu l'opportunité de travailler sur le corpus Arabic TreeBank ATB (Maamouri, 2004). Il s'agit d'un corpus contenant 120 transcriptions d'émissions d'actualité en arabe standard diffusées par différentes chaines arabes. Le corpus transcrit contient 51 080 mots annotés morpho-syntaxiquement et syntaxiquement. Pour créer un lexique en dialecte tunisien, nous avons essayé de construire en partant de l'ATB un lexique de traduction ASM-DT. Pour ce faire, nous avons adopté une méthode de transformation, du ASM vers le dialecte, basée sur les parties du discours des mots de l'ATB. Ceci permettra d'obtenir non seulement des dictionnaires bilingues mais aussi un ATB en tunisien utile pour des applications TALN (Figure 1). Nous expliquons dans ce travail les modèles de transformation ainsi que les structures des dictionnaires que nous avons définis pour les verbes et les différents outils syntaxiques de l'ATB.

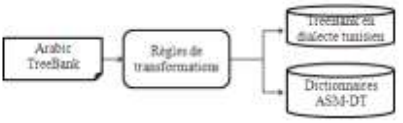


FIGURE 1 – Méthode pour la création de ressources en dialecte tunisien

3.2 Construction d'un lexique pour les verbes

Comme nous visons à adapter les outils ASM au dialecte tunisien, nous avons essayé de construire pour les verbes en DT les mêmes concepts que ceux d'ASM. En arabe, les principaux concepts verbaux sont:

1-Lemme: Il s'agit d'un concept fondamental dans l'analyse des textes. Les mots arabes peuvent être analysés comme étant une racine insérée dans un modèle constituant ainsi les lemmes de mots. Les verbes dans l'ATB sont présentés sous leur formes fléchies, nous avons extrait leurs lemmes et leurs racines en utilisant l'analyseur morphologique ELEXIR FM développé par (Smrž, 2007). Étant donné que nous sommes des locuteurs natifs du dialecte tunisien, nous avons construit manuellement à chacun des lemmes ASM des lemmes en dialecte. En résultat, nous avons constaté que 60% des verbes se comportent différemment en passant vers le dialecte ce qui prouve la différence entre le couple ASM-DT. Ainsi, ayant 1500 lemmes en DT et partant du fait que les verbes en ASM possèdent des schèmes décrivant leurs comportements morphologiques lors de la

conjugaison, nous avons cherché à attribuer des schèmes aux verbes en DT.

2-Pattern: Les patterns ou les schèmes en ASM sont des modèles avec différentes structures qui sont appliquées à la racine pour créer un lemme. Pour chaque racine, nous pouvons appliquer différents modèles pour avoir des lemmes avec des significations différentes. Le challenge dans la construction des schèmes pour les verbes dialectaux consiste à trouver des modèles similaires à ceux en ASM. Ainsi, en étudiant la morphologie des lemmes dialectaux, nous avons remarqué qu'il est possible d'attribuer aux lemmes en DT les mêmes modèles que ceux en ASM mais en définissant en plus d'autres modèles qui seront des schèmes fils pour les schèmes de base. En fait, ce processus a permis de distinguer 32 schèmes pour les verbes en DT, alors qu'il y avait 15 en ASM. Cela est dû à la richesse morphologique des lemmes dialectaux. Par exemple:

En ASM: le verbe $Arak/ شارك(forme au passé) yu$Arik/participerيشارك(forme au présent) ; et دافع/dAfaE (forme au passé) يدافع/yudAFiE/défendre(forme au présent) appartiennent au ASM-pattern-II: CvACvC (forme au passé)/yvCvACvC (forme au présent). Notons que les voyelles sont les mêmes pour les deux verbes. En passant vers le dialecte, les racines ainsi que les modèles de ces verbes restent les mêmes CvACvC/yvCACvC mais la voyelle de la seconde consonne n'est plus la même pour les deux verbes. Or en ASM, la marque de cette voyelle est un critère fondamental pour classer un verbe sous un pattern (Ouerhani, 2009) c'est pourquoi nous avons proposé de définir des sous-patterns pour le pattern II, et ce en divisant le pattern-II en pattern-II-i: CACiC/yvCACiC et pattern-II-a:Cacac/yvCACaC. Par conséquent, $Arak/yu$Arik qui devient en DT $Arik/yi$Arik/ va appartenir au pattern-II-i: CACIC/yiCACiC et dAfaE/yudAFiE qui devient en DT dAfaE/yidAFaE va appartenir au pattern-II-a: CACAC / yiCACaC. Donc, en adoptant ce raisonnement, nous avons réussi avec les verbes de l'ATB à définir des schèmes pour les verbes en DT.

3-Racine: Elle est la source fondamentale de toutes les formes des verbes arabes. La racine n'est pas un vrai mot, il s'agit plutôt d'une séquence de trois consonnes qui peut être trouvée dans tous les mots qui lui sont liés. La plupart des racines sont composées de trois lettres, très peu sont de quatre ou cinq consonnes. En dialecte tunisien, il n'existe pas encore de standard pour la définition de la racine. Pour cela, la construction de racine en dialecte n'est pas évidente, surtout quand le verbe change complètement de racine en passant du ASM vers le dialecte. En fait, pour définir une racine pour les verbes TUN, nous avons adopté une méthode déductive. En effet, la règle en ASM dit que racine + schème = lemme (1). Dans notre cas, nous avons déjà défini le lemme TUN et le schème TUN. En suivant la règle (1), l'extraction de la racine est rendue alors facile. Par exemple, nous avons classé le lemme إِسْتَنَّى /Aistan~aY/Attendre dans le schème AistaCCaC

Racine (?) + AistaCCaC = إِسْتَنَّى /Aistan~Y

En suivant (1), la racine est alors «ننى » [nnY]. En fait, nous pouvons dire que la définition des racines est une question problématique et qui pourrait admettre plus de discussion. D'après la démarche adoptée, c'est comme si, nous avons forcé la racine à être [nnY]. En effet, si nous classons إِسْتَنَّى /Aistann~aY sous le schème AiCtaCaC, la racine dans ce cas doit être سنن /snn. La racine peut être aussi quadrilatère سنني /snnY si nous classons إِسْتَنَّى /Aistann~aY sous le schème AiCCaCaC. Mais comme il n y'a pas de

standard, nous avons fait de notre mieux pour être le plus logique possible en définissant la racine dialectale.

3.3 Modélisation des concepts verbaux dialectaux dans le lexique ASM-DT

Les différentes transformations verbales décrites ci-dessus, sont modélisées et stockées dans un dictionnaire de verbes de la manière suivante: chaque bloc verbal ASM, contenant le lemme-ASM, schème-ASM et la racine-ASM, lui correspond respectivement un bloc DT contenant le lemme-TUN, la racine-TUN et le schème-TUN. La connaissance du bloc TUN nous permet de définir automatiquement les différentes formes fléchies du verbe TUN. La Figure 2 décrit la structure que nous avons définie pour stocker une unité verbale ASM-DT.

```
<DIC_TUN_VERBS_FORM>
  <LEXICAL-ENTRY POS="VERB">
<VERB ID-VERB="48">
    <ASM-LEMMA>
     <Headword-ASM>عَايَنَ</Headword-ASM>
      <Pattern>فاعل</Pattern>
      <Root-ASM>عين</Root-ASM>
      <Gloss lang= "fr" > Observer</Gloss>
    </ASM-LEMMA>
<TUN-VERB Sense= "1" >
   <Cat-Tun-Verb Category= "TUN--VERB--I--au--yi" />
<Root-Tun-Verb>شوف</Root-Tun-Verb>
<Conjug-Tun-Verb>
<TENSE>
<FORM Type= "IV" >
<VOICE Label="Active">
<Features Val_Number_Gender="1S">
<Verb_Conj>نشُوفْ</Verb_Conj>
<Struct-Deriv>Ø+شوف+ن</Struct-Deriv>
</Features>
</VOICE>
<VOICE Label="Passive">
      :::
</VOICE>
</Form >
<FORM Type= "PV" >
  <VOICE Label="Active">
       ::
</VOICE>
 <VOICE Label="Passive">
::::
</VOICE>
</Form >
<FORM Type= "CV" >
 <FeaturesVal_ Number _Gender="2S">
<Verb_Conj>شُوفْ</Verb_Conj>
      <Struct-Deriv>Ø+شوف+Ø</Struct-Deriv>
    </Features>
  </FORM>
 </TENSE>
 </Conjug-Tun-Verb
</TUN-VERB>
 </LEXICAL-ENTRY>
</DIC_TUN_VERBS_FORM>
```

FIGURE 2 – Structure de stockage d'une unité verbale

3.4 Règles de transformation pour la traduction des mots outils:

3.4.1 Transformation dépendante du contexte

Pour ce type de transformation, nous avons proposé de décrire les différents contextes dont peut dépendre un mot outil sous forme de règles. Nous désignons par une règle basée-contexte, le passage ASM-DT qui s'appuie sur des règles de transformation. En effet étant donnée un mot MK, on dit que la transformation de MK se base sur le contexte s'il donne une nouvelle traduction à chaque fois qu'on lui change le contexte. RT_k: $X + M + Y = TD_k$

X = $\sum_{i=1}^{m}$ Mj: POSj ; Y = $\sum_{i=1}^{n}$ Mi: POSi ; k varie de 1 à z ;

RT_k: Règle de transformation n°$_k$; POS : Partie de discours ; M: Mot outil ; TDk: Traduction n°$_k$.

Pour chaque mot outil, plusieurs configurations peuvent se présenter donnant à chaque fois une nouvelle traduction. La transformation d'un mot outil peut dépendre soit des mots qui le précèdent (X), soit qui le suivent (Y), soit des deux. Si aucun contexte ne se présente alors une traduction par défaut sera affectée au mot outil. Prenons l'exemple de la particule « حَتَّى »/HatY /pour que/ qui possède la POS: sub-conj dans l'ATB. Pour cette particule nous avons développé des règles conformément à trois contextes différents vus dans l'ATB.

1- HatY/حَتَّى + verb = باش/bA$ (TUN-particle) + TUN_verb (DIC-TUN-Verb)

2–HatY/حَتَّى + NEG_PART = bA$/باش(TUN-particle) + TUN_NEG_PART(DIC-TUN-NEG_PART).

Sinon

3- HatY/حَتَّى = HatY/حَتَّى (dans tous les autres contextes)

Le tableau 1 montre la manière dont on représente une règle dans le lexique. En effet, pour chaque mot outil nous avons défini un ensemble de contextes, chaque contexte contient une ou plusieurs configurations. La configuration décrit la position et la partie de discours du mot par rapport au mot outil. Chaque contexte lui correspond une traduction en tunisien. La traduction peut être soit directe soit indirecte c.à.d. elle fait appel à un autre dictionnaire de notre base lexicale (autre que celui du mot outil concerné).

Règle de transformation	HatY/حَتَّى + verbe = باش (TUN-particle) + TUN_verbe
Représentation de la règle dans le dictionnaire	<SUB_Conj ID="10"> <ASM-LEMMA>حَتَّى</ASM-LEMMA> <GLOSS lang="fr">Jusqu'à ce que / à</GLOSS> <CONTEXT ID="1"> <CONFIG ID= "1" Position="Après" POS="Verb" /> <TOKEN> <TUN ID="1">باش </TUN> <TUN ID="2"> </TUN> <TUN ID="3" DIC= « verbs » POS="verb" /> </TOKEN> </CONTEXT> …… <CONTEXT ID="3"> ….. </Sub_Conj>

TABLE 1 – Structure de stockage d'une transformation dépendante du contexte

3.4.2 Transformation syntaxique

Les transformations qui requièrent le changement de l'ordre syntaxique est une catégorie de transformation qui est aussi dépendante de contexte. Il s'agit de changer l'ordre des mots pour qu'ils aient un sens en dialecte. Dans notre travail, nous avons traité le niveau syntaxique au niveau de quelques groupes nominaux tels que:

ASM: كُتُب كَثِيرة /kutubun kavirap/ Noun + ADJ

DT: برشا كتُب/ bar$A ktub/ ADJ + Noun

L'intérêt ici, consiste à montrer la faisabilité de ce type de transformation et qu'on pourra intégrer dans notre base lexicale d'autres règles de ce type. Nous pourrons penser par exemple à changer les structures VSO (Verbe Sujet Object) en des structures SVO (Sujet Verbe Objet) puisqu'elles sont fréquemment utilisés en dialecte (Baccouche, 2003). Le tableau 2 illustre le stockage d'une règle contentant une transformation d'ordre syntaxique.

Transformation syntaxique	ASM: Noun+ ADJ -> DT: ADJ+Noun
Représentation de la règle dans le dictionnaire	<Noun-ASM ID="5"> <ASM-LEMMA> كُتُب </ASM-LEMMA> <GLOSS lang="fr">livres</GLOSS> <CONTEXT ID="1"> <CONFIGID="1"Position="Après"POS="ADJ" /> <TOKEN> <TUNID="1"DIC="ADJECTIVES" POS="ADJ" /> <TUN ID="2" /> <TUN ID="3">كتُب </TUN> </TOKEN> </CONTEXT>

TABLE 2 – Structure de stockage d'une transformation syntaxique

L'étude des différents contextes de mot outils nous a permis de développer 316 règles. Le tableau 3 montre le nombre de règles développés pour chaque mot outils.

	Préposition	Conjonction	Pseudo-verbe	Adverbe	Pronom	Particule	Interjection
Occurrence_ ASM	49924	36498	1505	1662	1642	6245	38
Mot différent_ASM	13	23	7	36	44	23	6
Nbre de règles	141	42	24	45	53	51	6

TABLE 3 – Statistique des règles de transformation développées

4 Génération automatique des corpus en dialecte tunisien

A fin d'assurer le recueil du maximum possible de ressources, nous avons développé un outil baptisé Tunisian Dialect Translator (TDT). Ce dernier est capable de générer automatiquement des textes en tunisien en exploitant le lexique bilingue développé et de l'enrichir. Le TDT fonctionne selon la démarche suivante :

1-Etiqueter morphosyntaxiquement un texte ASM: Chaque corpus textuel en ASM est analysé morphosyntaxiquement à l'aide de l'analyseur MADA (Morphological Analyser and disambiguisator of Arabic Dialect) (Habash, 2010). Il s'agit d'un outil multitâche: Il effectue à la fois la segmentation, la discrétisation, la lemmatisation, l'analyse morphologique et l'étiquetage morphosyntaxique. L'apport principal de cet outil est la désambigüisation.

2- Exploiter la base-lexicale ASM-DT: En se basant sur chaque partie du discours résultant de l'étiquetage de MADA, nous exploitons la base lexicale ASM-dialecte que nous avons développée et ce en créant pour chaque structure ASM sa traduction correspondante en DT.

3-Enrichir le lexique: le texte obtenu de l'étape précédente n'est pas toujours traduit parfaitement, vu que la base lexicale ne couvre pas tous les mots. Pour cela, dans le but d'améliorer la qualité de la traduction et d'enrichir davantage notre lexique, nous avons développé un module d'enrichissement semi automatique. Il permet de filtrer tous les mots ASM pour lesquels une traduction n'a pas été fournie. Ces mots sont intégrés d'une manière semi-automatique dans le lexique après avoir proposé les traductions correspondantes. La figure 3 illustre la démarche décrite.

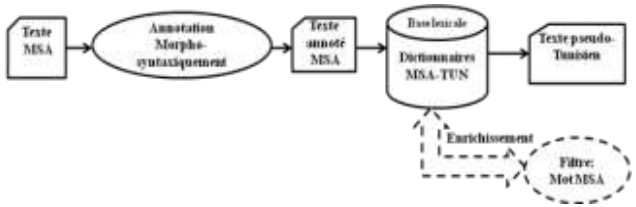


FIGURE 3 – Méthode pour la génération automatique de corpus en dialecte tunisien

5 Évaluation

Dans cette section, nous présentons une évaluation pour le lexique développé. En effet, nous avons demandé à des juges, qui sont des locuteurs natifs du dialecte tunisien, de proposer des traductions à des unités lexicales prises de notre dictionnaire.

5.1 Évaluation du lexique des verbes

Pour évaluer les verbes, nous avons proposé dans un premier temps à des juges de nous traduire un échantillon de verbes. En effet, parmi les 1500 verbes, nous avons pris aléatoirement un échantillon contenant 150 verbes. L'échantillon comporte 52 verbes qui ne changent pas du ASM vers le dialecte et 98 verbes qui changent complètement. Nous avons demandé par la suite à 47 juges de proposer des traductions à ces verbes. Les pourcentages calculés traduisent le pourcentage d'accord pour chaque verbe entre les traductions des juges et la traduction proposée dans notre lexique. Le tableau 4 représente les résultats obtenus.

Verbes	Inchangés	Changés	Total
Nombre de verbes	52	98	150
Accord	97,17%	63,21%	74,97%

TABLE 4 – Évaluation des verbes

La baisse de l'accord au niveau des verbes inchangés est due au fait que nous n'avons pas pris en compte l'aspect sémantique en faisant la traduction des verbes. En effet, un verbe peut avoir plusieurs sens selon la phrase où il se trouve.

5.2 Évaluation du lexique des mots outils

Puisque la traduction de la majorité des mots outils dépend du contexte, nous avons donné à 5 juges 89 phrases contenant 133 mots outils. Les mots outils se répètent parfois dans les phrases mais diffèrent du contexte. Nous avons demandé aux juges de traduire seulement les mots outils.

	2 juges	3 juges	4 juges	5 juges
Accord	72,69%	74,53%	71,34%	71,23%
Désaccord Total	18,79%	15,03%	14,28%.	12,03%

TABLE 5 – Évaluation des mots outils

Le tableau (5) donne les pourcentages d'accord entre les traductions des juges et celles de notre outil. La variation des pourcentages est due au fait que pour quelques mots outils, les juges ne s'accordent pas entre eux-mêmes. Le tableau présente aussi les

pourcentages de désaccord total entre les juges et le système. Le désaccord total se présente lorsqu'aucun juge ne donne une traduction similaire à celle donnée par le système. En augmentant le nombre de juges, le désaccord diminue ce qui prouve que notre base lexicale est capable de générer des traductions acceptables par plusieurs juges.

6 Conclusion

Dans cet article, nous avons décrit un processus de création de lexique et de génération de texte en dialecte tunisien dans le but de créer des corpus textuels pour entrainer un modèle de langage d'un système de reconnaissance. Ce processus s'est déroulé en deux phases. Dans la première, nous nous sommes focalisés sur la création d'une base bilingue ASM-DT en partant de l'Arabic TreeBank. Cette base a été exploitée aussi dans un travail d'adaptation de MAGEAD [Habash et al 2006] (Morphological Analyser and Generator of Arabic Dialect) au dialecte tunisien. Ceci est bien expliqué dans (Hamdi et al., 2013). Dans la deuxième phase, nous avons exploité cette base pour automatiser la tâche de la génération des textes en dialecte tunisien. L'orientation future de ce travail consiste à enrichir la base lexicale afin d'élargir la couverture lexicale dialectale. Nous envisageons aussi à proposer un modèle de langage pour les corpus dialectaux qu'on a obtenus et les évaluer par rapport à des transcriptions réelles. Des expériences en cours de réalisation sur le modèle de langage pour ces types de corpus ont montré que l'intégration de ces nouveaux corpus peut influer avantageusement sur le modèle de langage.

Remerciements

Je tiens à témoigner ma sincère reconnaissance à l'étudiante de master Mlle.Siwar benAyed qui m'a aidée à réaliser ce travail. Je remercie également M .Fréderic Bechet, Mme Mariem Ellouze, et Mme Lamia Belguith pour leurs remarques précieuses et leur participation au cheminement de ce travail au sein du laboratoire ANLP MIRACL et LIF Marseille .

Références

BACCOUCHE, T. (1994). L'emprunt En Arabe Moderne, Beit Elhikma Et Iblv, Tunis.

BACCOUCHE, T. (2003). La langue arabe: spécificités et évolution.

DIKI-KIDIRI, M. (2007). Comment assurer la présence d'une langue dans le cyberespace, UNESCO, Paris.

GRAJA, M., JAOUA, M. ET HADRICH BELGUITH, L. (2011). Building ontologies to understand spoken Tunisian dialect, *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, pp.23-32.

HABASH, N., RAMBOW, O., ROTH, R. (2009). MADA + TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR),* Cairo, Egypt.

HAMDI, A., BOUJELBANE, R., HABASH, N., NASR, A., Un système de traduction de verbes entre arabe standard et arabe dialectal par analyse morphologique profonde, *Actes de TALN2013 (Traitement automatique des langues naturelles)*, Nante, France.

NIMAAN, A., NOCERA, P. ET TPRRES-MORENO, JM. (2006). Boîte à outils TAL pour des langues peu informatisées : le cas du somali. *JADT* 2006 (*Journées internationales d'Analyse statistique des Données Textuelles)*. France, pp.694-701.

MAAMOURI,M. ET BIES, A. (2004). Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools, *Workshop on Computational Approaches to Arabic Script-based Languages*, COLING, Genève, Suisse.

SENG, S. (2010). Vers une modélisation statistique multi-niveau du langage, application aux langues peu dotées, thèse de doctorat, université de Grenoble, France.

SCHERRER,Y. (2008), Transducteurs à fenêtre glissante pour l'induction lexicale, *RECITAL*, Avignon, France.

SHAALAN, K., ABOUBAKR, HM., ET ZIEDAN,I. (2007). Transferring Egyptian Colloquial Dialect into Modern Standard Arabic. *RANLP (International Conference on Recent Advances in Natural Language Processing),* pp.525-529.Brovets, Bulgarie.

SMRŽ, O. (2007). Computational Approaches to Semitic Languages, ACL, Prague.

OUERHANI,B., (2009), Interférence entre le dialectal et le littéral en Tunisie: Le cas de la morphologie verbale, Synergies Tunisie n° 1, pp. 75-84, Tunisie.

Détection automatique des sessions de recherche par similarité des résultats provenant d'une collection de documents externe

Simon Leva[1] Nicolas Faessel[2]
(1) CLLE–ERSS : CNRS et Université de Toulouse (UMR 5263),
5 allées Antonio Machado, 31058 Toulouse Cedex 9
(2) IRIT : CNRS et Université de Toulouse (UMR 5505),
118 route de Narbonne, 31062 Toulouse Cedex 9
sleva@univ-tlse2.fr, nicolas.faessel@irit.fr

RÉSUMÉ

Les utilisateurs d'un système de recherche d'information mettent en œuvre des comportements de recherche complexes tels que la reformulation de requête et la recherche multitâche afin de satisfaire leurs besoins d'information. Ces comportements de recherche peuvent être observés à travers des journaux de requêtes, et constituent des indices permettant une meilleure compréhension des besoins des utilisateurs. Dans cette perspective, il est nécessaire de regrouper au sein d'une même session de recherche les requêtes reliées à un même besoin d'information. Nous proposons une méthode de détection automatique des sessions exploitant la collection de documents WIKIPÉDIA, basée sur la similarité des résultats renvoyés par l'interrogation de cette collection afin d'évaluer la similarité entre les requêtes. Cette méthode obtient de meilleures performances que les approches temporelle et lexicale traditionnellement employées pour la détection de sessions séquentielles, et peut être appliquée à la détection de sessions imbriquées. Ces expérimentations ont été réalisées sur des données provenant du portail *OpenEdition*.

ABSTRACT

Automatic search session detection exploiting results similarity from an external document collection

Search engines users apply complex search behaviours such as query reformulation and multitasking search to satisfy their information needs. These search behaviours may be observed through query logs, and constitute clues allowing a better understanding of users' needs. In this perspective, it is decisive to group queries related to the same information need into a unique search session. We propose an automatic session detection method exploiting the WIKIPEDIA documents collection, based on the similarity between the results returned for each query pair to estimate the similarity between queries. This method shows better performance than both temporal and lexical approaches traditionally used for successive session detection, and can be applied as well to multitasking search session detection. These experiments were conducted on a dataset originating from the *OpenEdition* Web portal.

MOTS-CLÉS : Recherche d'information, détection automatique de sessions de recherche, analyse de journal de requêtes.

KEYWORDS: Information retrieval, automatic search session detection, query log analysis.

1 Introduction

De plus en plus d'utilisateurs effectuent des recherches d'information sur le Web. Ils utilisent pour cela des moteurs de recherche, leur permettant d'exprimer leur besoin d'information sous la forme de requêtes constituées de mots-clés. Ces systèmes atteignent cependant leurs limites face à des requêtes comportant en moyenne deux ou trois mots-clés, n'exprimant pas un besoin d'information suffisamment explicite par rapport à l'ensemble des documents disponibles (Silverstein *et al.*, 1999). En particulier, les requêtes soumises par les utilisateurs tendent à être trop génériques ou trop spécifiques, nécessitant un certain nombre de reformulations avant d'obtenir un ensemble de documents pertinents (Downey *et al.*, 2007). Les requêtes d'un utilisateur sont donc rarement isolées, mais font essentiellement partie d'une session de recherche. Les sessions fournissent de nombreux indices sur l'objectif de l'utilisateur ou son expertise dans le domaine considéré, et constituent ainsi une unité qu'il peut être utile d'identifier en vue d'améliorer les performances d'un moteur de recherche.

Nous avons montré dans une précédente étude (Leva, 2013) que la segmentation d'un journal de requêtes en sessions de recherche n'est pas une tâche triviale pour des annotateurs humains, aboutissant à un taux d'accord modéré. Nous avons également observé que la réalisation de cette tâche a nécessité de la part des annotateurs une consultation de plusieurs ressources externes afin de pallier un manque de connaissances encyclopédiques, et ainsi permettre une prise de décision. Nous faisons donc l'hypothèse qu'il serait possible de développer une méthode de détection automatique des sessions basée sur la similarité entre les requêtes à partir de leur croisement avec une collection de documents.

Nous présentons dans cet article l'élaboration d'une méthode de détection de sessions exploitant les documents de la partie française de *Wikipédia* [1]. Cette méthode est évaluée sur une collection de référence construite à partir d'un journal de requêtes issu du portail *OpenEdition* [2], et les résultats obtenus sont comparés à deux méthodes de référence.

Dans une première section, nous présentons la notion de session à travers ses définitions et les différentes approches de détection automatique. Puis, nous présentons les données que nous avons utilisées en vue de nos expérimentations. Nous détaillons enfin les différentes méthodes de détection que nous avons mises en œuvre, avant d'en faire une évaluation.

2 État de l'art

Les journaux de requêtes *(query logs)* conservent une trace d'un certain nombre d'interactions entre des utilisateurs et un moteur de recherche (soumission et reformulation de requête, navigation sur les pages de résultats, consultation de documents...). Ces données permettent ainsi d'étudier les comportements de recherche des utilisateurs et fournissent des indices sur leurs besoins d'information. Dans cette perspective, la notion de *session de recherche* est centrale, entraînant le développement de diverses méthodes de détection automatique.

1. http://fr.wikipedia.org
2. http://www.openedition.org/

2.1 Définitions de la notion de session

Une session de recherche regroupe l'ensemble des requêtes soumises par un même utilisateur afin de satisfaire un même besoin d'information. Si cette idée de regrouper les différentes formulations d'un même besoin informationnel au sein d'une même unité est partagée dans les définitions de la notion de session, celles-ci font cependant l'objet de nombreuses variations. En effet, selon que leur structure soit envisagée de manière séquentielle ou imbriquée, les sessions vont comporter des caractéristiques différentes.

2.1.1 Structure séquentielle des sessions

L'une des premières définitions de la notion de session dans le cadre de l'étude d'un journal de requêtes a été proposée par (Silverstein *et al.*, 1999) :

> *A session is a series of queries by a single user made within a small range of time. A session is meant to capture a single user's attempt to fill a single information need.*

Ainsi, les requêtes appartenant à une même session se succèderaient dans l'ordre chronologique de leur soumission, aboutissant à une organisation des requêtes successives en séquences. L'une des implications de cette conception est que les sessions sont marquées par une longueur, que celle-ci soit exprimée en termes de nombre de requêtes ou d'unité de temps. D'une part, une session peut contenir une seule ou plusieurs requêtes (Gayo Avello, 2009). Dans ce dernier cas, la requête initiale est suivie d'une ou plusieurs reformulations (He *et al.*, 2002). D'autre part, au niveau temporel, la durée d'une session peut varier de moins d'une minute (Spink *et al.*, 2006), quelques minutes (He et Göker, 2000), à quelques heures (Spink *et al.*, 2006). Dans ces différents cas, la durée d'une session reste courte et inférieure à une journée. En effet, l'identification des utilisateurs dans un journal de requête se basant sur l'adresse IP, et celle-ci pouvant changer toutes les 24 heures, il est difficile de retrouver un utilisateur unique au-delà de cette période (Gayo Avello, 2009). Certains auteurs fixent ainsi une fenêtre temporelle de 24 heures sur les requêtes provenant d'une même adresse IP (Jansen *et al.*, 2007; Gayo Avello, 2009), correspondant à la notion d'*épisode de recherche*. Un épisode peut donc comporter une ou plusieurs sessions de recherche.

2.1.2 Structure imbriquée des sessions

Si la vision des sessions en tant que séquences de requêtes successives coïncide avec les enregistrements temporellement ordonnés constitués par les journaux de requêtes, elle ne reflète pas la complexité des parcours de recherche des utilisateurs. En effet, ces derniers peuvent mener une recherche simultanément sur plusieurs thèmes (par exemple à travers l'utilisation de plusieurs onglets dans leur navigateur), ou interrompre momentanément leur recherche en cours pour s'intéresser à un nouveau besoin d'information. Ce comportement, correspondant à une recherche multitâche *multitasking search*, peut se traduire par une alternance entre des requêtes visant chacune un besoin d'information distinct, et donc par des sessions imbriquées entre elles. Comme le montre l'étude de (Spink *et al.*, 2006), les recherches multitâches peuvent être très fréquentes dans certains environnements : dans un journal de requêtes du moteur *AltaVista*, respectivement 81 % et 91 % des séquences de 2 et 3 requêtes portent sur plusieurs thèmes à la fois. (Jones et Klinkner, 2008) envisagent ainsi que les requêtes liées à un même besoin d'information ne

sont pas nécessairement contiguës, mais peuvent s'intercaler avec des requêtes liées à un autre besoin d'information, donnant lieu à une imbrication entre sessions. Malgré cette autre manière d'envisager la structure des sessions, la notion d'épisode et son implication temporelle reste applicable, car liée à la problématique d'identification des utilisateurs.

2.2 Méthodes de détection automatique des sessions

Selon la structure des sessions adoptée, les méthodes de détection automatique font appel à des caractéristiques des sessions et des ressources différentes. Ces méthodes peuvent ainsi exploiter la durée des sessions, le contenu lexical des requêtes, et des sources de connaissance externes.

2.2.1 Méthode basée sur la durée des sessions

La première méthode de détection automatique des sessions à avoir été développée s'appuie sur la dimension temporelle des sessions, et envisage donc leur structure comme séquentielle. Cette méthode repose sur l'observation que plus la durée entre deux requêtes consécutives est longue, moins il est probable que ces requêtes renvoient à un même besoin d'information, et donc qu'elles appartiennent à une même session. Tout l'enjeu réside alors dans le choix d'un seuil temporel approprié fixant la durée maximale entre deux requêtes successives appartenant à la même session : 5 minutes (Silverstein *et al.*, 1999), 15 minutes (He et Göker, 2000), ou encore 30 minutes (Jansen *et al.*, 2007). Malgré sa forte utilisation due à sa simplicité de mise en œuvre, cette approche ne détecte ni les sessions très courtes résultant d'un changement soudain du besoin d'information, ni à l'inverse les sessions très longues au cours desquelles l'utilisateur peut effectuer des pauses importantes entre chaque requête. La prise en compte de ces cas nécessite en effet de s'appuyer sur d'autres indices de lien entre les requêtes.

2.2.2 Méthode basée sur le contenu lexical des requêtes

Afin de dépasser les limites de l'approche temporelle, une méthode de détection automatique exploitant le lien lexical entre les requêtes visant un même besoin d'information a été élaborée. L'hypothèse est alors que plus les requêtes ont un contenu lexical en commun, plus il est probable qu'elles appartiennent à une même session. La détection de ces liens lexicaux a principalement été envisagée à travers la tâche de détection des reformulations entre des requêtes successives. Plusieurs types de reformulation ont ainsi été définis (He *et al.*, 2002; Ozmutlu et Çavdur, 2005; Jansen *et al.*, 2007) : spécialisation (ajout d'un ou de plusieurs termes), généralisation (suppression d'un ou de plusieurs termes), reformulation (ajout et suppression d'un ou de plusieurs termes), etc. Si aucun de ces types de reformulation n'est identifié entre des requêtes successives, il est alors considéré que celles-ci ne sont pas lexicalement reliées, et appartiennent donc à des sessions différentes. Cette méthode a également été combinée avec la méthode temporelle, que ce soit à travers l'apprentissage automatique (He *et al.*, 2002; Ozmutlu et Çavdur, 2005) ou une interprétation géométrique (Gayo Avello, 2009). Néanmoins, l'approche lexicale possède deux principaux inconvénients : elle nécessite la présence d'au moins un mot commun entre les requêtes, et se heurte aux phénomènes de changement sémantique (synonymie, hyperonymie, hyponymie...). Le lien entre les requêtes visant un même besoin d'information

n'étant pas toujours lexicalement explicite mais pouvant être d'ordre sémantique, de nouvelles approches envisagent ainsi d'autres façons de détecter la similarité existant entre ces requêtes.

2.2.3 Méthode basée sur des sources de connaissance externes

Les approches temporelle et lexicale de détection automatique se basent sur une vision séquentielle des sessions, ne prenant une décision qu'à partir de la comparaison entre les requêtes successives d'un même utilisateur. De plus, ces approches ne permettent pas d'exploiter le lien de similarité souvent implicite entre les requêtes d'une même session. Plusieurs méthodes exploitent ainsi des sources de connaissance externes au journal de requête afin d'évaluer la similarité entre requêtes non plus de manière directe, mais à travers une représentation plus riche du contenu de ces requêtes. Ce niveau de représentation permet donc d'une part d'envisager la détection des sessions imbriquées, et d'autre part de prendre en compte toute la complexité de la tâche de détection des sessions.

(Jones et Klinkner, 2008) développent une approche basée sur un apprentissage supervisé exploitant des traits temporels, lexicaux, de cooccurrence des requêtes dans un journal plus étendu, et de similarité des requêtes avec les 50 premiers documents retournés en résultats. (Lucchese *et al.*, 2011) combinent deux mesures de similarité entre les requêtes : une similarité lexicale associant mesure de Jaccard et distance de Levenstein, ainsi qu'une similarité sémantique utilisant la mesure du cosinus sur une expansion des requêtes à l'aide des corpus WIKIPÉDIA et WIKTIONARY. Enfin, (Kramár et Bieliková, 2012) exploitent la similarité entre les métadonnées des documents pertinents cliqués pour chaque requête afin d'estimer la similarité entre chaque paire de requêtes. La pertinence des documents est ici estimée à l'aide d'un retour implicite, effectué par le système, des actions de l'utilisateur *(implicit feedback)*. Que ce soit au travers des documents constituant les résultats de la requête ou d'une collection de documents externe, ces approches présentent de meilleures performances que les approches lexicale et temporelle. En effet, elles permettent de prendre en considération la variété des comportements de recherche des utilisateurs, tant au niveau de la soumission de requêtes non contiguës qu'au niveau de la nature implicite du thème des requêtes. Cela pose également la question de l'évaluation des sessions imbriquées obtenues, présentant une perspective différente de celle des sessions séquentielles.

3 Données

Nous avons appliqué et évalué nos méthodes de détection automatique sur une collection de référence manuellement annotée en sessions provenant d'un journal de requêtes du portail *OpenEdition*. Nous présentons cet environnement de recherche ainsi que le journal de requêtes original avant de détailler la collection de référence utilisée.

3.1 Le portail *OpenEdition*

Le portail *OpenEdition* propose un libre accès à un ensemble de ressources électroniques dans le domaine des sciences humaines et sociales. Développé et dirigé par le Centre pour l'édition électronique ouverte (Cléo), il se compose de trois plateformes dont chacune est dédiée à une

ressource électronique spécifique : *Revues.org* diffuse 363 revues et 16 collections de livres, *Calenda* recense plus de 21 000 évènements scientifiques en lettres et en sciences humaines et sociales, tandis qu'*Hypotheses.org* héberge 613 blogs et carnets de recherche.

Plusieurs points d'entrée permettent d'effectuer une recherche dans cet environnement varié. D'une part, un moteur de recherche principal est accessible sur la page d'accueil du portail *OpenEdition* et de la plateforme *Revues.org*. D'autre part, une recherche peut également se faire directement à partir du moteur de recherche situé sur le site d'une revue associée à *Revues.org*. Dans ces deux situations, les résultats sont présentés dans une interface commune.

3.2 Journal de requêtes initial

Nous avons exploité un journal de requêtes provenant du portail *OpenEdition* contenant une collection de 1 057 471 requêtes soumises par 227 302 utilisateurs durant la période du 07 avril 2010 au 1er février 2012. La langue principale des requêtes est le français, mais certaines sont également en anglais ou en espagnol. À la différence des requêtes soumises à un moteur de recherche généraliste, l'environnement d'*OpenEdition* se distingue par le fait que les requêtes proviennent essentiellement d'acteurs du monde académique et ciblent des revues, des évènements ou des blogs dans le domaine des sciences humaines et sociales.

La construction de ce journal de requêtes a nécessité la mise en œuvre de plusieurs traitements afin de ne conserver que les informations les plus fiables à partir du journal d'accès *(access log)* original. En particulier, les données ont subi plusieurs opérations de nettoyage et de filtrage visant à éliminer les informations inexploitables (requêtes soumises par des robots d'indexation, suites de signes de ponctuation, etc.). Les requêtes ont ensuite été regroupées par adresse IP et classées par ordre chronologique. Le journal de requêtes finalement obtenu comporte un identifiant pour chaque utilisateur – correspondant à l'adresse IP anonymisée –, la date et l'heure de soumission de chaque requête, ainsi que les requêtes soumises.

3.3 Collection de référence

Dans une précédente étude (Leva, 2013), nous avons constitué une collection de référence à partir d'un échantillon du journal de requêtes *OpenEdition* comportant 947 requêtes soumises par 216 utilisateurs. Cette collection a été manuellement annotée en sessions afin de servir de référence à la fois pour l'évaluation de méthodes de détection automatique des sessions et pour des études sur les types de reformulations effectuées par les utilisateurs. L'ensemble des requêtes a été automatiquement segmenté en 349 épisodes de recherche, correspondant pour chaque utilisateur à l'ensemble de ses requêtes soumises en une journée au plus. La durée entre chaque requête successive est connue, mais cette information n'est pas présentée aux annotateurs. Trois annotateurs non spécialistes dans les domaines représentés par les documents ont été chargés de regrouper les requêtes de chaque épisode de recherche en une ou plusieurs sessions.

La tâche d'annotation des sessions a fait l'objet d'un guide d'annotation. Pour chaque épisode de recherche d'un utilisateur, les annotateurs devaient observer l'ensemble des requêtes soumises avant de prendre une décision et d'attribuer une session à chaque requête à travers un identifiant numérique unique. La collection de référence contient ainsi des sessions imbriquées. Afin d'identifier les requêtes visant un même besoin d'information, les annotateurs pouvaient

s'appuyer sur plusieurs indices, à la fois textuels, sémantiques, et de proximité thématique, ou s'appuyer à défaut sur des ressources externes permettant de pallier un manque de connaissances encyclopédiques. Un accord inter-annotateur a été évalué à l'aide du coefficient Kappa, le taux d'accord variant de modéré (0,47 et 0,57) à bon (0,61). La collection de référence finale contient ainsi 406 sessions pour les 947 requêtes initiales, résultant des annotations faisant l'objet d'un accord entre au moins deux annotateurs.

4 Méthodes de détection automatique de sessions

Nous proposons d'utiliser différentes méthodes de détection automatiques de sessions : une méthode temporelle, une méthode lexicale se basant sur les indices lexicaux des requêtes pour identifier si une requête est une reformulation de la requête précédente, et enfin une méthode exploitant la collection de documents WIKIPÉDIA. Les deux premières méthodes, qui sont des méthodes provenant de la littérature, nous serviront de références pour la détection de ruptures de sessions dans le contexte d'identification de sessions de recherche séquentielles, que nous proposons dans la section 5.1.

4.1 Exploitation d'un seuil temporel

À partir d'un seuil fixant la durée maximale entre deux requêtes successives pouvant appartenir à la même session, la méthode temporelle détecte les ruptures (durée entre les requêtes supérieure au seuil) et les continuités (durée entre les requêtes inférieure au seuil) de session au sein de chaque épisode de recherche d'un même utilisateur. Chaque requête se voit ainsi attribuer un identifiant de session unique au sein d'un même épisode. Ce type d'approche n'envisageant les requêtes que d'un point de vue séquentiel, les sessions imbriquées ne sont pas repérées, et les identifiants de session sont constamment incrémentés pour chaque nouvelle session d'un épisode.

4.2 Exploitation des liens lexicaux pour la détection de reformulation

Dans leur étude, (Jansen *et al.*, 2007) proposent de détecter différents types de reformulation en utilisant les liens lexicaux entre deux requêtes. L'idée est de compter le nombre de mots communs à deux requêtes, et de déterminer ensuite, grâce à la longueur de chacune, si l'utilisateur a spécifié sa requête, ou bien l'a généralisée, etc. Dans le cas de sessions séquentielles, sans imbrications, on peut faire l'hypothèse que si une reformulation est détectée entre deux requêtes consécutives, c'est que celles-ci font partie d'une même session. Sinon, la dernière requête représente une nouvelle session : il y a eu une rupture de la session de recherche précédente. Dans le cadre de sessions imbriquées, la seule exploitation des liens sémantiques est plus délicate : la temporalité est implicitement utilisée dans la reformulation, car une requête ne peut être une reformulation que d'une requête antérieure. Ainsi, pour détecter les sessions imbriquées, il faut comparer toutes les requêtes entre elles en préservant leur ordre temporel.

4.3 Exploitation de la similarité des requêtes à l'aide de *Wikipédia*

Comme nous l'avons mentionné dans la section 2, l'utilisation de la temporalité et la détection des différents types de reformulation ne sont pas suffisantes pour la détection de sessions séquentielles, et encore moins dans le cadre des sessions imbriquées. Ainsi, nous proposons l'utilisation d'une source d'information externe qui est une version locale de *Wikipédia* en français, datant du 28 octobre 2012 [3]. Ce corpus a été indexé dans un moteur de recherche (*Terrier* [4]), permettant ainsi son interrogation avec les requêtes des épisodes de recherche.

Contrairement aux approches exploitant la sémantique des documents provenant d'une source externe (comme par exemple la *wikification* proposée par (Lucchese *et al.*, 2011; Kramár et Bieliková, 2012)), nous exploitons la liste des documents, ainsi que leur score, renvoyés par le système de recherche. Ainsi, les listes de résultats obtenus pour chaque requête d'un même épisode sont comparées, afin d'estimer si les requêtes forment une session de recherche.

Soit E un épisode de recherche. Soit Q_E l'ensemble des requêtes de l'épisode de recherche. Soit R_{q_a} l'ensemble des résultats de la requête $q_a \in Q_E$. R_{q_a} est constitué d'un ensemble de documents pondérés. Le poids des documents correspond aux scores obtenus par ces documents lors de l'interrogation dans un moteur de recherche. Le nombre de documents renvoyés par le moteur de recherche est fixé à un maximum de 1 000. Ce paramètre n'a pas été étudié dans le présent article, bien que le nombre de documents réellement pertinent dans *Wikipédia* pour une requête donnée est probablement plus faible. R_{q_a} peut être représenté comme un vecteur de dimension M, où M correspond à l'espace des documents du corpus, et dont les coordonnées sont données par le score des documents appartenant à M, obtenus pour la requête q_a. On peut représenter l'ensemble des documents renvoyés pour une requête comme un vecteur $\overrightarrow{v}_a = (s_{1,q_a}, s_{2,q_a}, \ldots, s_{i,q_a}, \ldots, s_{M,q_a})$, où s_{i,q_a} correspond au score du document i pour la requête q_a.

La similarité entre deux requêtes q_a et q_b est donnée par le cosinus de l'angle des vecteurs de documents répondant à ces deux requêtes :

$$sim(v_a, v_b) = \frac{\overrightarrow{v}_a \bullet \overrightarrow{v}_b}{|\overrightarrow{v}_a| \times |\overrightarrow{v}_b|} \quad (1)$$

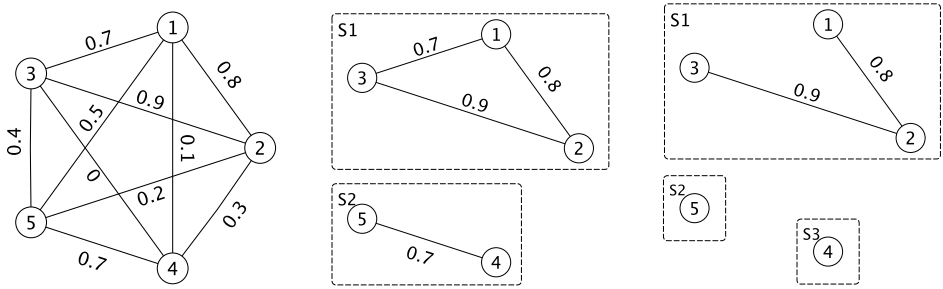
Un épisode peut être représenté comme un graphe valué non orienté complet $\mathcal{G}_{\mathcal{E}}(N, A)$, dont les nœuds N correspondent aux requêtes, et les arêtes A correspondent à la similarité entre deux requêtes (figure 1a). On peut définir le graphe de sessions $\mathcal{G}_{\mathcal{S}}(N, A')$, comme le sous graphe partiel potentiellement non connexe de $\mathcal{G}_{\mathcal{E}}$ dont l'ensemble des arêtes A' ont une valeur supérieure à un seuil t, dont chaque composante connexe forme une session de recherche.

Les figure 1b et 1c représentent deux graphes de sessions, déterminés respectivement pour un seuil de similarité $t = 0,7$ et $t = 0,8$.

5 Analyse des résultats

Nous proposons une analyse des résultats de notre méthode de détection automatique (cf. section 4.3) en prenant en compte à la fois une perspective séquentielle et imbriquée sur les

3. `http://dumps.wikimedia.org/frwiki/20121028/frwiki-20121028-pages-articles.xml.bz2`
4. `http://terrier.org`



(a) Graphe de l'épisode $\mathscr{G}_{\mathscr{E}}$ (b) Graphe de sessions pour le seuil 0.7 (c) Graphe de sessions pour le seuil 0.8

FIGURE 1 – Représentation en graphe des épisodes et sessions de recherche

sessions détectées. Au niveau de l'évaluation des sessions séquentielles, nous effectuons une comparaison entre notre méthode et celles basées sur une approche temporelle et lexicale.

5.1 Évaluation des sessions séquentielles

Nous avons exploité la collection de référence OPENEDITION pour l'évaluation des trois méthodes de détection automatique des sessions ainsi que pour le réglage des paramètres internes de chacune de ces méthodes. Dans le cadre de sessions séquentielles, cette évaluation peut être réalisée à l'aide des mesures de précision, rappel et F-mesure, appliquées au nombre de ruptures et de continuations de sessions détectées par chaque système. Une rupture peut être détectée entre deux requêtes consécutives si, dans la méthode temporelle, le temps entre les deux requête est supérieur à un délai donné, et dans le cas de la méthode lexicale, aucun cas de reformulation n'est identifié. Ce type d'évaluation ne permet donc pas de refléter entièrement les performances d'un système dans le cadre de la détection de sessions imbriquées.

5.1.1 Mesures d'évaluation

Les équations suivantes présentent les mesures d'évaluation proposées par (Gayo Avello, 2009) dans le cadre d'une comparaison avec une référence manuelle. La précision P correspond au nombre de ruptures de session présentant un accord entre le système et la référence par rapport au nombre de ruptures de session détectées par le système. Le rappel R correspond au nombre de ruptures de session présentant un accord entre le système et la référence par rapport au nombre de ruptures de session de la référence. La F-mesure F permet de pondérer rappel et précision. Nous avons automatiquement annoté notre collection de référence en ruptures et continuations de session, les cas où deux requêtes successives d'un même épisode possèdent un même identifiant de session correspondant à une continuation, les cas contraires à une rupture. Nous avons également supprimé les cas triviaux pour lesquels les systèmes n'ont aucune décision à prendre (la première requête de chaque épisode et un épisode ne contenant qu'une seule requête constituent toujours une rupture de session) afin d'évaluer leurs performances réelles,

réduisant à 598 le nombre de requêtes de la collection de référence initiale.

$$P = \frac{N_{RuptureCorrecte}}{N_{RuptureSysteme}} \qquad R = \frac{N_{RuptureCorrecte}}{N_{RuptureReference}} \qquad F = \frac{2PR}{P+R} \tag{2}$$

5.1.2 Méthode temporelle

Précision	0,31
Rappel	0,31
F-mesure	0,31

(a) Mesures d'évaluation

		Rupt.	Cont.	
Réf.	Rupt.	22	48	70
	Cont.	49	479	528
		71	527	598

(b) Matrice de confusion

TABLE 1 – Résultats de l'évaluation de la méthode temporelle.

Le tableau 1a présente les performances de la méthode temporelle pour une durée maximale des sessions de 640 secondes. Après avoir testé cette méthode avec des valeurs de seuil allant de 10 à 5 120 secondes, ce seuil offre en effet les meilleurs résultats sur notre collection.

Le tableau 1b présente la matrice de confusion des résultats de la méthode temporelle. Cette méthode possède une efficacité de 84 %, correspondant à la proportion des vrais positifs (22) et des vrais négatifs (479) par rapport à l'ensemble des cas traités (598). Nous observons que la méthode temporelle entraîne autant de faux positifs (tableau 2, utilisateur 18) que de faux négatifs (tableau 2, utilisateur 32). Cela est directement lié à l'incapacité de cette méthode de s'adapter à la durée propre à chaque session. Ainsi, il s'est écoulé 4 436 secondes entre les requêtes de l'utilisateur 18 qui sont explicitement liées au niveau lexical, et 315 secondes entre les requêtes de l'utilisateur 32 qui portent sur des thèmes distincts.

Util.	Requête	Réf.	Syst.
18	`loup`	1	1
	`Le monde agricole confronté au loup`	1	2
32	`Après la catastrophe`	1	1
	`Recherches sociologiques et anthropologiques`	2	1

TABLE 2 – Exemple de faux positif et de faux négatif induits par la méthode temporelle.

5.1.3 Méthode lexicale

Le tableau 3a présente les performances de la méthode lexicale sur notre collection de référence.

Le tableau 3b présente la matrice de confusion des résultats de la méthode lexicale, montrant une efficacité de 62 %. La méthode lexicale n'entraîne aucun faux négatif. En effet, de par son fonctionnement, cette méthode considère par défaut qu'il existe une rupture de session entre les requêtes si aucun type de reformulation n'est détecté, ce qui explique le taux de rappel de 1

Précision	0,24
Rappel	1
F-mesure	0,38

(a) Mesures d'évaluation

		Rupt.	Cont.	
Réf.	Rupt.	70	0	70
	Cont.	225	303	528
		295	303	598

(b) Matrice de confusion

TABLE 3 – Résultats de l'évaluation de la méthode lexicale.

observé. Cette méthode est donc sensible au contenu lexical des requêtes, et les faux positifs proviennent de l'absence de mots communs entre les requêtes, des fautes de frappe non palliées par la distance d'édition, ainsi que de la non détection des liens sémantiques entre les requêtes. C'est le cas dans le tableau 4 pour l'utilisateur 35, dont l'ensemble des requêtes renvoie au thème des produits éclaircissants pour la peau.

Util.	Requête	Réf.	Syst.
35	`éclaircissants`	1	1
	`peau claire`	1	2
	`tshoko`	1	3
	`maquillage afrique`	1	4
	`dépigmentation`	1	5

TABLE 4 – Exemple de faux positif induit par la méthode lexicale.

5.1.4 Méthode basée sur *Wikipédia*

Précision	0,31
Rappel	0,8
F-mesure	0,45

(a) Mesures d'évaluation

		Rupt.	Cont.	
Réf.	Rupt.	56	14	70
	Cont.	124	404	528
		180	418	598

(b) Matrice de confusion

TABLE 5 – Résultats de l'évaluation de la méthode basée sur *Wikipédia*.

Le tableau 5a présente les performances de la méthode basée sur *Wikipédia* pour un seuil de similarité entre requêtes fixé à 0,005. Après avoir testé cette méthode avec des valeurs de seuil allant de $1 \cdot 10^{-5}$ à 0,5, ce seuil offre en effet les meilleurs résultats sur notre collection.

Le tableau 5b présente la matrice de confusion des résultats de la méthode basée sur *Wikipédia*, montrant une efficacité de 77 %. Cette méthode génère presque neuf fois plus de faux positifs que de faux négatifs. Le cas de l'utilisateur 7 dans le tableau 6 est un exemple de faux négatif. Ces cas correspondent à des requêtes contenant peu de termes ou des termes génériques, facilitant la découverte de liens thématiques entre elles, et constituent également des cas ambigus pour les annotateurs. Il aurait ainsi été possible de regrouper l'ensemble des requêtes de l'utilisateur 7 au sein d'une même session ayant pour thème la finance. Les perspectives d'améliorations sont

donc faibles pour ce type d'erreurs. En revanche, les faux positifs sont dûs à des fautes de frappe et pour l'essentiel aux limites de la collection de documents de *Wikipédia*, qui contiennent peu ou pas d'information concernant certaines requêtes très caractéristiques de l'environnement de recherche *OpenEdition*. C'est le cas de l'utilisateur 71, qui effectue une recherche portant sur un auteur de revue spécifique de la plateforme *Revues.org*. Une solution serait alors d'exploiter la collection de documents d'*OpenEdition* conjointement à ceux de *Wikipédia* de manière à avoir une couverture à la fois spécifique et générique sur les sujets introduis par les requêtes.

Nous pouvons néanmoins observer qu'au niveau de la détection des ruptures de sessions, la méthode basée sur *Wikipédia* donne une précision égale à celle de la méthode temporelle $(0,31)$ tout en améliorant le rappel $(0,8)$. Les performances globales de cette approche, représentées par une F-mesure de $0,45$, sont meilleures que celles des approches temporelle et lexicale implémentées.

Util.	Requête	Réf.	Syst.
71	`philippe dasseto`	2	2
	`dasseto`	2	3
	`Dasseto`	2	4
7	`bank`	1	1
	`bank crisis`	1	1
	`China party`	2	1
	`china financial`	2	1
	`bank`	1	1

TABLE 6 – Exemple de faux positif et de faux négatif induits par la méthode basée sur *Wikipédia*.

5.2 Évaluation des sessions imbriquées

Les données sur lesquelles nous avons effectué nos expérimentations contiennent des sessions de recherche imbriquées. Cette expérimentation permet de valider, non plus la détection des points de rupture, mais bien les sessions renvoyées par notre système par rapport aux sessions de référence annotées manuellement.

Nous utilisons ici certaines des métriques définies par (Lucchese *et al.*, 2011). Ces mesures sont l'index de Rand (Rand, 1971) et l'index de Jaccard (Jaccard, 1901), qui considèrent des paires de requêtes et permettent de vérifier la cohérence de répartition de ces paires entre les sessions système et les sessions de référence d'un même épisode de recherche.

On considère f_{11} le nombre de paires qui sont dans une même session calculée et dans une même session de référence, f_{00} le nombre de paires reparties dans des sessions calculées différentes, ainsi que dans des sessions de référence, f_{01} le nombre de paires qui sont dans une même session calculée mais dans des sessions de références différentes, f_{10} le nombre de paires qui sont des sessions calculées différentes, mais dans une même session de référence.

L'index de Rand est défini comme suit :

$$R = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

et l'index de Jaccard :

$$J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

Seuil	0,00001	0,00005	0,0001	0,0005	0,001	0,005	0,01
Rand	0,78556	0,78556	0,78556	**0,78796**	0,78646	0,75840	0,72841
Jaccard	0,72588	0,72588	0,72588	**0,72829**	0,72228	0,66834	0,62587

TABLE 7 – Similarité des sessions système par rapport aux sessions de référence.

Les résultats du tableau 7 montrent que le meilleur seuil de similarité pour la génération automatique des sessions est égal à 0,0005. Ce seuil, très bas, s'explique par le fait que la liste des résultats obtenus pour chaque requête est très sensible aux différents types de reformulation. En effet, l'ajout ou la suppression d'un mot dans la requête peut changer totalement les résultats renvoyés par le moteur de recherche utilisé. Ainsi, bien que l'utilisation de la similarité entre les résultats des requêtes puisse aider à détecter des sessions imbriquées, il semble évident que le seuil de détection optimal de ces sessions est dépendant de notre jeu de données.

6 Conclusion

Nous avons proposé une méthode de détection des sessions de recherche basée sur l'utilisation d'une ressource externe. Cette méthode, exploitant les résultats d'un moteur de recherche sur la collection de documents de *Wikipédia*, a été validée dans le cas de sessions de recherche séquentielles. Nous avons également proposé une extension dans le cadre de la recherche de sessions imbriquées. Une perspective à court terme est de valider la détection de sessions imbriquées par rapport aux autres approches de la littérature utilisées dans ce cadre.

Les résultats préliminaires obtenus sont encourageants, et nous envisageons d'étudier la combinaison des trois approches, à savoir temporelle, lexicale, et basée sur l'utilisation de *Wikipédia*. En effet, il nous semble que ces approches sont complémentaires : l'approche utilisant la similarité des requêtes calculée au moyen d'une ressource externe est très sensible aux reformulations identifiées dans l'approche lexicale. Une idée serait que lorsque la reformulation est identifiée de manière triviale, celle-ci prédomine dans la détection de session. Si ce n'est pas le cas, le système peut utiliser la similarité pour détecter les sessions. Une autre piste que nous envisageons concerne l'exploitation de la collection de documents de la plateforme *OpenEdition*, sans doute mieux adaptée au contenu des requêtes de notre journal.

Remerciements

Ce travail s'inscrit dans le projet ANR CAAS (*Contextual Analysis and Adaptive Search*, programme Contint) coordonné par Josiane Mothe (IRIT), faisant l'objet d'un partenariat entre l'Institut de Recherche en Informatique de Toulouse (IRIT), le Laboratoire Informatique d'Avignon (LIA) et l'Équipe de Recherche en Syntaxe et Sémantique du laboratoire Cognition, Langue, Langage, Ergonomie (CLLE–ERSS). Nous adressons également nos plus vifs remerciements à Marin Dacos

et l'équipe du Centre pour l'édition électronique ouverte (Cléo) pour leur collaboration et la mise à disposition des données du portail *OpenEdition*.

Références

DOWNEY, D., DUMAIS, S. et HORVITZ, E. (2007). Models of searching and browsing : Languages, studies, and applications. *In Proc. IJCAI*, pages 2740–2747.

Gayo AVELLO, D. (2009). A Survey on Session Detection Methods in Query Logs and a Proposal for Future Evaluation. *Information Sciences*, 179(12):1822–1843.

HE, D. et GÖKER, A. (2000). Detecting Session Boundaries from Web User Logs. *In Proceedings of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research*, pages 57–66.

HE, D., GÖKER, A. et HARPER, D. J. (2002). Combining Evidence for Automatic Web Session Identification. *Information Processing and Management*, 38(5):727–742.

JACCARD, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.

JANSEN, B. J., SPINK, A., BLAKELY, C. et KOSHMAN, S. (2007). Defining a Session on Web Search Engines. *Journal of the American Society for Information Science and Technology*, 58(6):862–871.

JONES, R. et KLINKNER, K. L. (2008). Beyond the Session Timeout : Automatic Hierarchical Segmentation of Search Topics in Query Logs. *In Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 699–708.

KRAMÁR, T. et BIELIKOVÁ, M. (2012). Detecting Search Sessions Using Document Metadata and Implicit Feedback. *In Proceedings of the WSCD 2012 Workshop on Web Search Click Data*.

LEVA, S. (2013). Les sessions de recherche comme contexte des requêtes. *In Actes de l'atelier Contextualisation de Messages Courts – 13e Conférence Francophone sur l'Extraction et la Gestion des Connaissances (EGC'13)*, pages 1–12.

LUCCHESE, C., ORLANDO, S., PEREGO, R., SILVESTRI, F. et TOLOMEI, G. (2011). Identifying Task-Based Sessions in Search Engine Query Logs. *In Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 277–286.

OZMUTLU, H. C. et ÇAVDUR, F. (2005). Application of Automatic Topic Identification on Excite Web Search Engine Data Logs. *Information Processing and Management*, 41(5):1243–1262.

RAND, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):pp. 846–850.

SILVERSTEIN, C., MARAIS, H., HENZINGER, M. et MORICZ, M. (1999). Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum*, 33(1):6–12.

SPINK, A., PARK, M., JANSEN, B. J. et PEDERSEN, J. (2006). Multitasking During Web Search Sessions. *Information Processing and Management*, 42(1):264–275.

Une approche mixte morpho-syntaxique et statistique pour la reconnaissance d'entités nommées en langue chinoise

Zhen Wang[1,2]
(1) GEOLSemantics, 32, rue Brancion, 75015, Paris
(2) INALCO, ERTIM, 2 rue de Lille, 75343, Paris
zhen.wang@geolsemantics.com

RÉSUMÉ

Cet article présente une approche mixte, morpho-syntaxique et statistique, pour la reconnaissance d'entités nommées en langue chinoise dans un système d'extraction automatique d'information. Le processus se divise principalement en trois étapes : la première génère des noms propres potentiels à l'aide de règles morphologiques ; la deuxième utilise un modèle de langue afin de sélectionner le meilleur résultat ; la troisième effectue la reconnaissance d'entités nommées grâce à une analyse syntaxique locale. Cette dernière permet une reconnaissance automatique d'entités nommées plus pertinente et plus complète.

ABSTRACT

A Mixed Morpho-Syntactic and Statistical Approach to Chinese Named Entity Recognition

This paper presents a morpho-syntactic and statistical approach for Chinese named entity recognition which is a part of an automatic system for information extraction. The process is divided into three steps : first, the generation of possible proper nouns is based on morphological rules; second a language model is used to select the best result, and last, a local syntactic parsing performs the named entity recognition. Syntactic parsing makes named entity recognition more relevant and more complete.

MOTS-CLÉS : Reconnaissance de noms propres, Reconnaissance d'entités nommées, Traitement automatique du chinois, Extraction d'information, Analyse syntaxique

KEYWORDS : Propre noun recogition, Named entity recognition (NER), Chinese Natural Language Processing, Information extraction, Syntactic parsing.

1 Introduction

La reconnaissance d'entités nommées (EN) joue un rôle crucial dans l'extraction de connaissances, les systèmes de question/réponse, la traduction et les résumés automatiques, ainsi que dans l'indexation inter-lingue. Cette tâche a pour objectif de déterminer les frontières d'une entité nommée, et de lui attribuer un type.

La définition d'une « entité nommée » varie selon les systèmes et l'utilisation qui en est faite. La campagne d'évaluation *Automatic Content Extraction* (ACE) en 2007 a étendu la définition traditionnelle des entités (ACE 2007) en ajoutant aux types habituels

(personne, organisme, lieu, expressions numériques) les types véhicule et arme. Dans le programme Quaero (Rosset et al., 2011), la définition des entités nommées a également été étendue en prenant en compte de nouveaux types (tels que les civilisations et les fonctions), et des expressions ne contenant pas de nom propre.

La définition des entités nommées que nous allons employer ici s'approche de celles d'ACE et de Quaero. Nous traitons donc les entités nommées de type : personne, organisme, lieu, expressions numériques (dates, heures, quantités, mesures, nombres), avec ou sans nom propre. Dans cet article, nous nous concentrons sur la présentation du traitement des entités nommées des type personne, organisme et lieu, en chinois, qui contiennent au moins un nom propre.

Cet article propose une approche mixte morpho-syntaxique et statistique pour la reconnaissance d'entités nommées en langue chinoise. Les méthodes existantes utilisées pour la reconnaissance d'EN en langue chinoise seront résumées dans la section 2. Les composants de notre approche seront exposés en section 3. Puis, nous présenterons dans la section 4 l'évaluation de notre système sur un corpus. Enfin, nous énoncerons des perspectives d'amélioration dans la section 5.

2 État de l'art

Les particularités de la langue chinoise rendent la reconnaissance d'EN plus difficile que pour les langues occidentales. En effet, l'un des premiers problèmes rencontrés est l'absence de séparateurs de mots. Par exemple, la phrase « Je vais à Paris. » se traduit ainsi en chinois « 我去巴黎。 ». Une étape de segmentation en mots est effectuée avant la reconnaissance d'EN. Cependant, la segmentation est très souvent ambiguë, ce qui multiplie les hypothèses de découpage. Par exemple, une suite de caractères comme « 的确切 » peut être découpée en « 的确/切 » (sûrement / couper) ou « 的/确切 ». (de / exactitude). Ceci rend difficile la détermination des frontières des mots, notamment dans le cas de mots inconnus, qui sont le plus souvent des noms propres (Sun et al., 2009). À cela, on peut ajouter le manque d'information typographique. Par exemple, il n'existe pas de différenciation entre majuscule et minuscule, qui constitue un critère de reconnaissance efficace pour les noms propres en français.

Les premiers travaux sur la reconnaissance d'EN en langue chinoise ont débuté au début des années 1990 (Sun et al., 2010). Les méthodes basées sur des règles (Wang et al., 1992) ont autant été employées que les méthodes statistiques (Sproat et al., 1990), telles que le modèle de Markov Caché (Liu et al., 2005, Wang et al., 2012) et les champs aléatoires conditionnels (*Conditional Random Fields*) (Chen et al., 2006, Mao et al., 2008). Les deux types de méthode ont chacun des avantages ainsi que des inconvénients. En effet, les méthodes basées sur des règles s'appuient sur des dictionnaires ou/et des règles linguistiques élaborés par des experts de la langue. Elles permettent d'obtenir une bonne précision pour certains cas, mais le processus de construction est long. De plus, il est difficile d'inclure tous les cas linguistiques existants. De ce fait, la portabilité de ces méthodes est faible. Les méthodes statistiques, quant à elles, s'appuient sur un corpus d'apprentissage. La construction d'un système est rapide mais la pertinence de ces méthodes dépend de la taille et du contenu du corpus d'apprentissage. D'autres recherches (Chen et al., 2000, Cao et al., 2002) se fondent sur une approche qui combine

les deux types de méthodes afin de profiter des avantages et de pallier les inconvénients de chaque approche. Nous nous plaçons dans cette dernière catégorie.

3 Notre approche

Notre module de reconnaissance d'entités nommées est basé sur un système d'automates à états finis pondérés. Ces automates effectuent l'analyse morphologique et l'analyse syntaxique des textes. Un automate particulier sert à la désambiguïsation. Il utilise un corpus étiqueté afin de créer un modèle de langue.

Nous utilisons les notions d'annonceur (McDonald , 1993) et de déclencheur , très importantes pour la reconnaissance des noms propres et des EN. En effet, le déclencheur peut être un mot ou une catégorie, qui permet de provoquer la détection d'un nom propre. Les déclencheurs doivent pouvoir être définis par une liste finie de mots ou catégories afin de pouvoir être intégrés dans les règles. Par exemple, nous avons désigné les noms de famille, qui représentent environ 131 mots pour les noms de personne chinois, comme mots déclencheurs pour la détection de prénoms potentiels. Quant aux annonceurs de nom propre, ce sont des mots qui sont suivis ou précédés par un nom propre faisant partie d'une entité nommée. Ces annonceurs de nom propre sont souvent utilisés comme annonceurs d'entité nommée. Il peut s'agir de mots qui désignent un métier, le titre d'une personne, un type de lieu, d'organisation ou de produit. Par exemple, le nom de famille «WANG» est le mot déclencheur permettant la reconnaissance du prénom «Zhen», et «Mademoiselle» peut être un annonceur de nom propre qui se situe au début d'une entité nommée de type personne comme «Mademoiselle Wang Zhen». Que ce soit le nom de famille «WANG» ou l'annonceur «Mademoiselle», l'un ou l'autre peut être utilisé comme mot déclencheur pour la reconnaissance de l'entité nommée «Mademoiselle Wang Zhen».

Dans notre corpus d'apprentissage, seuls les annonceurs immédiatement suivis ou précédés d'un nom propre ont été étiquetés avec la catégorie « annp », afin de faciliter la désambiguïsation de la phrase. Nous attribuons éventuellement une propriété à la catégorie « annp » afin de distinguer sa position par rapport au nom propre, il s'agit d' « antérieur » ou « postérieur ». Ceci permet aussi de mieux repérer les noms propres potentiels. Les autres potentiels annonceurs sont étiquetés comme tels juste avant l'analyse syntaxique, une fois que la désambiguïsation a été effectuée grâce aux catégories positionnelles. Ils vont permettre de détecter des structures syntaxiques particulières, de reconnaître des EN aux structures plus complexes et éventuellement de déterminer les types des EN. Par exemple, dans l'entité nommée « 胡主席 » (le président Hu), « 胡 » (hu) est étiqueté comme nom propre et « 主席 » (président) comme annonceur postérieur. En revanche, dans l'entité nommée « 中国石油大学 » (L'Université du pétrole de Chine) , puisque le nom propre « 中国 » (Chine) est suivi immédiatement par le nom « 石油 » (pétrole) mais pas par l'annonceur « 大学 » (Université), celui-ci n'est pas étiqueté comme annonceur dans la meilleure hypothèse de catégorisation après désambiguïsation (voir plus bas), mais le sera lors de l'analyse syntaxique.

3.1 Architecture

Notre procédure (figure 1) débute par une étape de tokenization qui permet de

reconnaître les caractères latins, et de les étiqueter de façon simple (seulement deux catégories sont utilisées). Ensuite, une segmentation en mots est effectuée à l'aide de dictionnaires. Elle est suivie par une étape de reconnaissance et de normalisation d'expressions numériques. Plusieurs hypothèses de segmentation du texte sont produites à l'issue de cette étape. Dans le texte, chaque mot est associé à une ou plusieurs catégories.

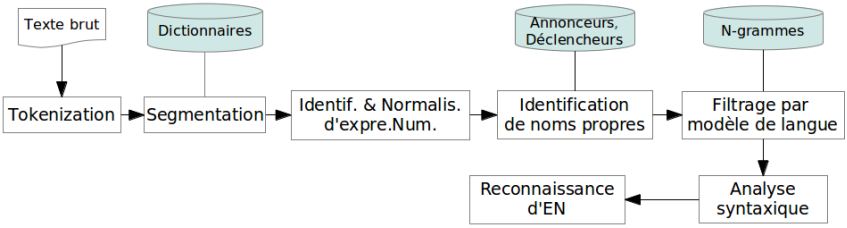


FIGURE 1: Procédure de reconnaissance d'entités nommées

Puis, l'identification de noms propres potentiels à l'aide de règles linguistiques et de ressources linguistiques (liste d'annonceurs, etc.) est réalisée. Celle-ci génère plusieurs hypothèses de segmentation et de catégorisation supplémentaires. Ensuite, un modèle de langue permet de désambiguïser et de sélectionner la meilleure segmentation. Enfin, la reconnaissance et le typage d'entités nommées est réalisée juste après l'analyse syntaxique.

3.2 Automates morphologiques

La construction morpho-syntaxique des noms propres en chinois est très variée. En effet, un nom propre peut être composé de n'importe quels caractères et a une longueur variable. Afin de tenir compte de ces phénomènes, lors du découpage et de la consultation des dictionnaires, nous ajoutons aux hypothèses d'interprétation des mots, des noms propres potentiels, obtenus grâce à des règles linguistiques.

La langue chinoise a peu de formes fléchies. Mais certaines catégories peuvent aider à exclure des mots qui ne peuvent pas entrer dans la composition des noms propres, tels que certaines dates, les interjections, etc. De ce fait, nous avons choisi de procéder à la reconnaissance des noms propres après l'étape de segmentation et de reconnaissance d'expressions numériques.

Les automates morphologiques (Eilenberg, 1974) sont des transducteurs qui contiennent des opérations telles que la composition et la concaténation. Ils permettent de traiter à la fois les caractéristiques communes à tous les noms propres, mais aussi leurs différences. Par exemple, en chinois, les noms propres de personnes chinois ont des particularités par rapport aux noms propres étrangers. C'est pour cette raison que nous avons décidé de traiter les noms propres de personnes chinois différemment. Par ailleurs, les noms propres transcrits de type personne, organisation ou produit ont des caractéristiques

communes. À ce stade, nous les traitons ensemble sans différencier leur type. De plus, nous avons pris en compte la relation entre un nom propre et son annonceur. C'est un autre critère sur lequel les automates morphologiques se basent pour identifier les noms propres.

Par conséquent, nous avons divisé la reconnaissance de noms propres en deux phases : l'identification des noms propres de personnes chinois, et l'identification des noms propres étrangers (personnes, lieux, organismes ou de type inconnu).

Notons que, dans cet article, nous n'avons pas décrit l'identification de noms propres en écriture latine, puisque les expressions en écriture latine sont traitées avec une méthode relativement simple, basée sur la reconnaissance typographique. Par exemple, une suite de caractères en écriture latine contenant une arobase « @ » est identifiée comme un nom propre, ainsi que celles commençant par un majuscule.

3.2.1 Noms de personne chinois

Les noms de personne chinois ont des particularités morphologiques. Tout d'abord, un nom de personne chinois ne possède qu'entre 2 et 4 caractères : il peut contenir un nom de famille de 1 à 2 caractères et un prénom de 1 à 2 caractères. Ensuite, le nom de famille précède toujours le prénom. Enfin, le nombre de noms de famille en chinois est limité, et les cent noms de famille les plus fréquents concernent 87% de la population chinoise (De La Robertie, 2005). Cependant, l'ambiguïté de segmentation entraîne souvent la confusion entre nom de famille et prénom.

Après avoir analysé les noms de personne chinois mal reconnus, nous pouvons lister les causes suivantes : 1) un caractère du prénom est inconnu ; 2) le deuxième caractère du prénom est ambigu avec un autre mot : « 薄/熙来/自 » (Bo / Xilai / à partir de) ou « 薄/熙/来自 » (Bo / Xi / vient de); 3) le premier caractère du prénom est un annonceur spécial : « 刘/庄 » (Liu / Zhuang) est le nom d'une personne ou « 刘/庄 » (Liu / Village) est le nom d'un village dont le nom est Liu, ce qui pose aussi une difficulté pour le typage des entités nommées; 4) le prénom ou le nom de famille font partie d'un nom commun, comme par exemple dans 2) ; 5) le prénom est un nom commun : « 李/建国 » (Li / Jiangguo ou Li / fonder le pays) ; 6) le nom de personne est un nom commun : « 汪/洋 » (Wan / Yang) ou « 汪 洋 » (Océan). Par conséquent, la reconnaissance de noms de personne chinois revient le plus souvent à une résolution des ambiguïtés de segmentation.

Notre automate utilise les noms de famille comme déclencheurs pour détecter des prénoms potentiels. Ces prénoms sont sélectionnés par leur catégorie et par leur nombre de caractères. Par exemple, nous pouvons exclure des prénoms potentiels les expressions numériques, notamment les dates ainsi que les expressions quantitatives, et les mots qui ont plus de 2 caractères.

La détection de noms propres potentiels s'effectue après la consultation du dictionnaire, lorsque la phrase a déjà plusieurs hypothèses de segmentation et d'étiquetage. Les noms propres potentiels sont identifiés à l'aide des règles de construction de noms propres. Par exemple, pour la phrase « 王 珍 去 北 京 » (Wang Zhen va à Beijing), nous obtenons plusieurs hypothèses de catégorisation après la consultation de dictionnaire(voir figure 2). En utilisant le nom de famille « 王 » (Wang) comme déclencheur, des possibilités ont

été ajoutées après la détection de noms propres (voir figure 3).

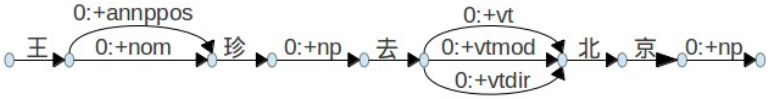


FIGURE 2: Transducteur de l'exemple "Wang Zhen va à Beijing" après la consultation de dictionnaire

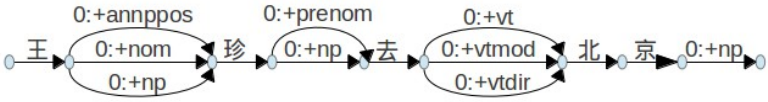


FIGURE 3:Transducteur de l'exemple "Wang Zhen va à Beijing" après la détection de nom de personne

Les exemples montrent que les automates morphologiques génèrent d'avantage d'hypothèses de segmentation et de catégorisation pour une phrase donnée. Ces hypothèses sont par ailleurs pondérées grâce aux règles linguistiques ce qui permettra d'effectuer une désambiguïsation à l'aide d'un modèle de langue, comme présenté dans la section 3.3.

3.2.2 Noms propres transcrits

À l'inverse des noms de personne chinois, les noms propres étrangers, notamment européens, sont transcrits en écriture chinoise d'après la phonétique du nom d'origine. Des caractères spécifiques sont utilisés pour la composition de ces noms propres. Les caractères utilisés sont peu présents dans la composition de mots communs par rapport aux autres caractères de même prononciation. Ceci réduit la possibilité de former un mot commun à l'intérieur d'un nom propre transcrit. Autrement dit, après notre segmentation, un nom propre étranger est séparé en plusieurs caractères individuels. Par exemple, la transcription du nom de personne « Nicolas Sarkozy » en écriture chinoise est « 尼古拉·萨科齐 », et le résultat de sa segmentation est « 尼/古/拉/·/萨/科/齐 », sans ambiguïté. La reconnaissance de ces noms propres transcrits consiste donc à regrouper les caractères qui les composent. De plus, les caractères utilisés dans la transcription appartiennent généralement à un ensemble limité.

Notre automate utilise une liste de caractères de transcription comme déclencheur afin de détecter des noms propres étrangers potentiels. Prenons l'exemple de « Nicolas Sarkozy », les caractères de sa translittération en écriture chinoise sont dans la liste. Notre automate les repère, puis les regroupe en deux noms propres potentiels, « 尼古拉 » et « 萨科齐 ». Grâce à la fonction conventionnelle du point de séparation « · », le prénom et le nom de famille seront identifiés.

Dans le cas où un nom propre transcrit ne contient que des caractères hors liste qui ont été étiquetés en +unk[1], nous avons mis en place un automate qui permet de prendre en compte ces caractères, le nombre de caractères, ainsi que le contexte de ce nom propre potentiel. Quand une chaîne de caractères n'est composée que de caractères transcrits, ceux-ci sont regroupés en un mot et nous considérons que c'est un nom propre. En observant dans un dictionnaire de noms de personnes du monde (Xinhua News Agency, 1993, 650 milles noms propres transcrits), nous avons constaté que le nombre de caractères du nom propre transcrit dépasse rarement 8. Cette caractéristique a été appliquée dans les règles. Le nom propre peut être précédé ou suivi par un annonceur. Le déclencheur de ce type de nom propre est souvent un caractère inconnu non latin du dictionnaire utilisé. Par exemple, l'un des résultats de l'analyse de « 巴拉圭 » (Paraguay) est « 巴+unk/ 拉+vt /圭+unk». Puisque « 拉 » est un caractère qui peut entrer dans la composition d'un nom propre transcrit, et que les caractères à sa gauche ainsi qu'à sa droite sont des mots inconnus, nous pouvons les regrouper pour former un nom propre potentiel.

3.3 Automates statistiques

Les automates statistiques se composent de dictionnaires pondérés, de règles linguistiques pondérées, et d'un modèle de langue. Ils permettent d'attribuer et de calculer la probabilité d'une suite de catégories pour une phrase entrée. L'attribution de la pondération est effectuée au moment de l'application des automates de segmentation, de reconnaissance de noms propres et de lissage. Quant aux noms propres, les pondérations sont attribuées à l'aide de la fréquence du nom de famille dans le corpus d'apprentissage, et à l'aide de règles linguistiques. Par exemple, le caractère « 王 » individuel a plus de chances d'être un nom de famille (Wang) qu'un nom commun (roi).

Le dictionnaire utilisé pour la segmentation (voir un extrait figure 4) contient 119 859 couples (mot, catégorie), dont 4 526 mots de catégorie « nom propre ». Un couple (mot, catégorie) peut être associé à un poids. Les couples non présents ou présents une seule fois dans le corpus sont sans pondération. Comme montré sur la figure 4, la première colonne est un mot en écriture chinoise, la deuxième colonne est la catégorie grammaticale du mot, la troisième colonne est le poids associé à ce couple (mot, catégorie). Ce poids représente le nombre d'occurrences du couple (mot, catégorie) dans le corpus d'apprentissage. Il est calculé par la formule suivante : $poids(w_i, cat_j) = -\log (\Sigma(w_i, cat_j))$, où $\Sigma(w_i, cat_j)$ désigne le nombre d'occurrences du mot w_i avec la catégorie cat_j dans le corpus d'apprentissage.

[1]+unk : caractère/mot inconnu

```
1 躺 +vt-2.19722457733622
2 制陶者 +nom  -0
3 决赛权 +nom  -0
4 叫停 +vt     -0.693147180559945
5 直属 +adjdet-2.19722457733622
6 免税 +adv    -0.693147180559945
7 音乐界 +nom  -0
8 李枝盈 +np   -0
```

FIGURE 4: – Exemples de couples (mot, catégorie) pondérés.

Le modèle de langue utilise des trigrammes de catégories morphosyntaxiques. Par exemple, « annppos » est la catégorie d'annonceur de nom propre postérieur. Ils sont établis à partir du corpus étiqueté du LDC (Chinese Treebank 6.0, 2007), et permettent d'attribuer une pondération aux séquences de catégories (figure 5), afin de calculer la catégorie la plus probable d'un mot en contexte. Par exemple, un nom propre est plus probable après un verbe qu'après un adverbe, après un annonceur qu'après un adjectif. Nous avons modifié le corpus LDC avec un jeu de catégories morphosyntaxiques défini par notre équipe afin d'optimiser la désambiguïsation. Par exemple, la catégorie « +vtmod » qui désigne le verbe de modalité a été ajoutée afin de mieux désambiguïser le nom et le verbe.

```
2 `+1pronpers'      `+vtmod'    `+1pronpers'      -1.79175946922805
3 `+nom'      `+num'      `+virgule'  -3.09104245335832
4 `+np' `+etc'      `+virgule'  -1.79175946922805
5 `+<#>'      `+adj'      `+specifnom'      -1.38629436111989
6 `+nomdir'   `+vt' `+pointfin' -5.04985600724954
7 `+nom'      `+pard'     `+parg'     -1.94591014905531
8 `+vtmod'    `+nom'      `+virgule'  -2.63905732961526
9 `+parg'     `+date'     `+conjcoord'      -1.38629436111989
```

FIGURE 5 – Exemples des trigrammes de catégories.

Le modèle de langue est appliqué sur les textes étiquetés. Les probabilités des différents chemins possibles sont alors calculées afin de résoudre les ambiguïtés de segmentation et de catégorisation. Après l'application du modèle de langue, nous ne conservons que la solution la plus probable.

3.4 Automates syntaxiques et reconnaissance d'EN

Les automates syntaxiques établissent des relations de dépendance typées entre les mots, en s'appuyant surtout sur leur catégorie et sur leurs propriétés. Ils mettent en évidence les liens (attribut, appositif ou quantitatif) entre les mots au sein des groupes nominaux, permettant ensuite d'identifier une entité nommée, même lorsque son annonceur est éloigné du nom propre. Nous avons spécifié quelque types de relations afin de mieux repérer les entités nommées. Par exemple, ENpers désigne les relations entre un nom de

famille et un prénom, ENrel entre un annonceur et un nom propre, ATT désigne les relations de type attribut entre un annonceur et un nom ou un adjectif, QUN désigne les relations de type quantitatif entre un chiffre et un nom comme 3人(3 personnes).

Une fois que les relations syntaxiques au sein des groupes nominaux ont été repérées, la dernière phase de la reconnaissance des entités nommées commence. Elle consiste à regrouper les relations syntaxiques (ENrel, ENpers, ATT, etc) qui composent l'entité nommée, et à typer celle-ci. Un automate repère dans un premier temps, un nom propre ou un annonceur dans la phrase donnée. Un autre automate parcourt toutes les relations qui ont comme tête ce nom propre ou cet annonceur en prenant en compte les types des relations et la position du nom propre ou de l'annonceur dans celles-ci. Nous les récupérons jusqu'à rencontrer un nom propre, un annonceur, ou bien une frontière de groupe nominal. Certains types de relations ne peuvent pas faire partie d'une entité nommée, comme les relations entre un sujet et un verbe, mais aussi les relations entre un prénom et une date, ce qui est aussi une condition d'arrêt du parcours.

La figure 6 présente un exemple d'analyse syntaxique de la phrase « 国家主席习近平同志访问法国。 » (Le président de l'État, le camarade Xi Jinping, visite la France). L'entité nommée reconnue dans cette phrase est « le président de l'État, le camarade Xi Jinping » qui inclut tous les composants liés par les relations de type ENpers et ceux liés par la relation ATT, liée au mot « Président ».

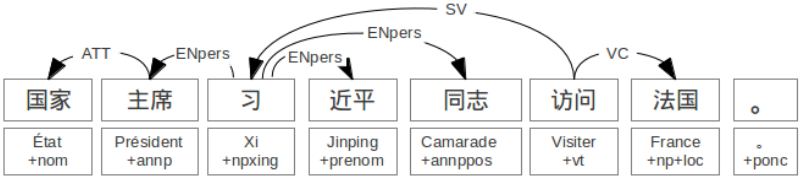


FIGURE 6 – Analyse syntaxique (exemple 1)

En revanche, dans la phrase « 摄影师拍摄阿拉伯海美景。 » (Le photographe photographie le beau paysage de la Mer d'Arabie)(voir la figure 7) , l'entité nommée reconnue est « Mer d'Arabie », mais n'inclut pas la relation entre l'annonceur postérieur « Mer » et le nom « beau paysage », car le nom « beau paysage » gouverne « Mer », mais n'est pas un annonceur.

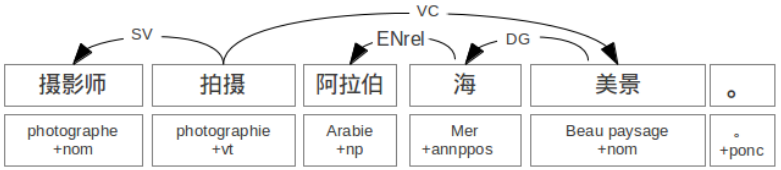


FIGURE 7 – Analyse syntaxique (exemple 2)

Le typage des entités nommées s'effectue après le regroupement de leurs composants. Le type de l'entité nommée est déterminé grâce au type de l'annonceur, qui est lui-même

déjà déterminé dans les dictionnaires. Par ailleurs, le type des entités nommées sans annonceur est déterminé grâce à la présence d'un nom de famille, et le type des entités nommées sans annonceur, ni nom de famille reçoit un type spécial « unk » (inconnu). Lorsque l'entité nommée contient plusieurs annonceurs, c'est le type du dernier qui détermine le type de cette entité nommée. Par exemple, l'entité nommée « 北京(Beijing +np+loc)市(Ville +annppos+loc[2])人民(peuple +nom)政府(gouvernement +annppos+org[3]) 网(net +annppos+prod[4]) » (le site web du gouvernement populaire de la ville de Beijing), est typée comme « produit », comme nous l'indique le type du dernier annonceur, 网(net +annppos +prod). Dans le cas d'entités nommées composées d'une seule unité comme « 北 京 »(Beijing), le type peut être indiqué directement dans le dictionnaire. Enfin, certaines entités nommées peuvent ne pas être typées à ce niveau, faute d'informations sémantiques. Nous utilisons alors le type spécial, « unk »(inconnu).

4 Évaluation

4.1 Corpus et calcul des scores

Pendant la deuxième campagne internationale SIGHAN[5] (*Second International Chinese Word Segmentation Bakeoff*), un corpus élaboré par l'Université de Pékin a été utilisé pour l'évaluation de la segmentation en chinois simplifié. Ce corpus est composé d'articles variés issus de journaux d'information, et est en chinois simplifié, la langue visée par notre traitement. Nous avons décidé d'annoter manuellement les entités nommées au sein d'un extrait[6] segmenté de ce corpus pour une première évaluation. 3 215 entités nommées ont été annotées, dont 1 055 de type personne, 1 565 de type lieu et 595 de type organisme.

Pour évaluer notre système, nous avons choisi d'utiliser le *slot error rate (SER)* (Makhoul et al., 1999) qui permet de mieux identifier les types d'erreurs. La formule est : SER = (I + D + TF + 0.5 * (T + F))/R ; un nombre plus petit indique une performance meilleure. La formule prend en compte :

- Insertion (I) : EN dans le test mais pas dans la référence ;
- Suppression (D) : EN dans la référence mais pas dans le test ;
- Type et frontière (TF) : EN dans le test avec erreurs de type et de frontière ;
- Type (T) : EN dans le test avec erreur de type ;
- Frontière (F): EN dans le test avec erreur de frontière ;
- Référence (R) : EN présentes dans la référence.

Par exemple, pour «国家主席习近平同志访问法国。» (Le président de l'État, le camarade Xi Jinping, visite la France), la référence est <pers>国家主席习近平同志</pers> 访问 <loc>法国</loc> où deux entités nommées ont été annotées. Supposons que nous ayons

[2]+loc : une propriété d'un mot qui signifie que le mot a un trait sémantique «lieu»
[3]+org : une propriété d'un mot qui signifie que le mot a un trait sémantique « organisme »
[4]+prod : une propriété d'un mot qui signifie que le mot a un trait sémantique « produit »
[5]Site web du SIGHAN : http://sighan.cs.uchicago.edu/
[6]Cet extrait du corpus annoté est disponible sous conditions d'utilisation.

les deux hypothèses suivantes :

1. 国家主席 <pers>习近平同志</pers> 访问 <loc>法国</loc>.
2. 国家主席 <pers>习近平同志</pers> 访问 <org>法国</org>.

L'hypothèse 1 contient une erreur de frontière pour l'EN de personne, donc le SER est 0.5*1/2=0,25. L'hypothèse 2 contient une erreur de frontière pour l'EN de personne et une erreur de type pour l'EN de lieu, donc le SER est (0.5*(1+1))/2=0,50.

4.2 Résultats et observations

Nous avons appliqué notre système sur cet extrait de corpus et obtenu un SER de 0,2589, 2 633 sur 3 215 entités nommées ont été correctement identifiées et typées. Le nombre d'erreurs par type est présenté dans le tableau ci-dessous (Table 1). ENpers désigne une entité nommée de type personne, ENloc est de type lieu, ENorg de type organisme et ENunk est le type inconnu.

	ENpers	ENloc	ENorg	ENunk	Total
Suppression	141	88	56	-----	285
Insertion	125	88	19	104	336
Type	5	35	3	0	43
Frontière	85	39	4	-----	128
Type + Frontière	25	39	62	0	126
Total	381	289	144	104	918
Total d'EN correctes	799	1 364	470	-----	2 633

TABLE 1 – Évaluation SER en chiffres

Certaines entités nommées de type personne (ENpers) ne sont par repérées à cause de l'absence de certains noms de famille qu'elles contiennent dans la liste des déclencheurs. Ces noms de famille ne sont pas intégrés dans la liste des déclencheurs à cause de leur faible présence dans le corpus, et de leur faible usage dans la vie courante. L'intégration de ces noms de famille doit être effectuée avec beaucoup de précautions. Elle nécessite plus de temps. Certaines autres entités nommées, telles que les surnoms, les noms de personne japonais, les abréviations d'organisme et des noms propres de lieux n'ont pas été identifiés à cause de leur structure morphosyntaxique très particulière.

Les entités nommées inconnues contiennent principalement des chiffres et des expressions en écriture latine. Ces chiffres sont écrits de manière inhabituelle. Par exemple : dans « 19·1% » (normalement écrit 19.1%), le point de séparation « · » est considéré dans nos règles comme un composant important pour identifier des entités

nommées. Dans notre système, une chaîne de caractères en écriture latine commençant par une majuscule est identifiée comme un nom propre. Certains de ces noms propres sont cependant des entités nommées de produit qui ne sont pas annotées dans le corpus de référence actuel.

Les entités nommées incorrectes détectées montrent des détections erronées de noms propres. Nous allons affiner l'utilisation de listes de déclencheurs en vue d'éviter cette détection erronée. Pour ce faire, la structure phonétique des noms propres transcrits ainsi que la structure interne des noms propres seront étudiées, afin d'intégrer nos nouvelles observations aux règles. Les erreurs de frontières et de type+frontière soulignent le problème de segmentation. Les entités nommées effacées et les erreurs de typage et de type+frontière révèlent des problèmes de catégorisation. Les entités nommées supprimées sont celles qui ne contiennent pas de nom propre ni d'annonceur, et qui ne sont pas repérables par un déclencheur, telles que des noms d'organisations ainsi que des produits.

5 Perspectives

Le résultat de l'évaluation est encourageant, 82 % des entités nommées ont été bien identifiées. Par la suite, nous allons également affiner la reconnaissance des entités nommées afin de traiter les entités nommées imbriquées les unes dans les autres. En effet, pour une exploitation d'entités nommées en vue de l'extraction d'information, nous souhaitons, pour certains cas, extraire plusieurs entités nommées à partir d'une même chaîne de caractères, afin de conserver des informations importantes. Par exemple, pour « 比利时(Belgique +np +loc)使馆(Ambassade +annppos +org) », l'entité nommée doit être l'Ambassade de Belgique, mais dans « 巴黎(Paris +np +loc)歌剧院(Opéra +annppos +org) », il est préférable d'extraire à la fois « Opéra de Paris » et « Paris », étant donné que « Paris » est le lieu où se trouve l'Opéra, contrairement à « Belgique » qui n'est pas le lieu où se trouve l'ambassade, et devrait être extrait et typé de manière différente, car « Belgique » dans « l'ambassade de Belgique » a davantage le sens d'organisation que de lieu.

Remerciements

Nous tenons à remercier l'Agence Nationale de la Recherche, projet portant la référence ANR-09-CSOSG-08-01, pour l'aide qu'elle nous a apportée pour mener à bien ce travail.

Références

ACE 2007. *The Automatic Content Extraction 2007 Evaluation Plan, Evaluation of the Detection and Recognition of ACE Entities, Value, Temporal Expressions, Relations, and Events*

Cao Wenjie, Zong Chengqing. *Chinese person name identification based on rules and statistics*. In : The 3th International Symposium on chinese spoken language processing, 2002

Chen Keh-Jiann and Chert Chao-jan. Knowledge Extraction for Identification of Chinese

Organization Names. In : *Second Chinese Language Processing Workshop, Association for Computational Linguistics,* 2000, pages 15-21

Chen W., Zhang Y., & Isahara H.. Chinese Named Entity Recognition with Conditional Random Fields. In : *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, Association for Computational Linguistics,* 2006, pages 118-121

De La Robertie Pierre. *Le nom propre en chinois*. *Essai de morphosyntaxe*. In : CORELA – Numéros thématiques le traitement lexicographique des noms propres. *Publié en ligne le 02 décembre 2005*

Eilenberg Samuel. 1974 *Automata, Languages, and Machines*. Volume A. Academic Press, San Diego.

Liu, F., Zhao, J., Lv, B., Bo, X., & Yu, H. Product Named Entity Recognition Based on Hierarchical Hidden Morkov Model. In : *Proceedings of the Forth SIGHAN Workshop on Chinese Language Processing*, 2005

Makhoul J., Kubala F., Schwartz, R. et Weischedel R. (1999). Performance measures for information extraction. In *Darpa broadcast news workshop*.

Mao Xinnian, Dong Yuan, He Saike, Wang Haila, Bao Sencheng. Chinese Word Segmentation and Named Entity Recognition Based on Conditional Random Fields. In *Sixth SIGHAN Workshop of Chinese Language Processing, 2008*

McDonald David D. Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In *Corpus Processing for Lexical Acquisition, 1993, pages 61-67*

Rosset Sophie, Grouin Cyril, Zweigenbaum Pierre. *Entités nommées structurées : guide d'annotation Quaero*. Notes et Documents LIMSI N° : 2011-04, Septembre, 2011

Sproat, Richard. and Shih, Chilin. A statistic method for finding word boundaries in Chinese text. In : *Computer Processing of Chinese & Oriental Languages,* 1990*, Vol 4*, pages 336-351

Sun Tieli, Liu Yanji, 中文分词技术的研究现状与困难 [The State of the art and difficulties in Automatic Chinese word segmentation]. In 信息技术 *[Information Technology], 2009, Vol.7* [en chinois]

Sun Zhen, Wang Huilin, 命名实体识别研究进展综述 [Overview on the advance of the research on named entity recognition]. In 现代图书情报技术 *[New technology of library and information service]*, 2010, No.7, pages 42-47 [en chinois]

Propre Names And Translation Service, Xinhua News Agency. *Names of the world's peoples – A comprehensive dictionary of names in roman-chinese.* Published by Chian Translation & Publishing Corporation, 1993

Wang, L.-J.; Li, W.-C. & Chang, C.-H.. Recognizing Unregistered Names for Mandarin Word Identification. In : *Proceedings of 14th COLING,* 1992, pages 1239-1243

Wang Longyue, Li Shuoo, Wong Derek F., Chao Lidia, A Joint Chinese Named Entity Recognition and Disambiguation System. In *The second CIPS-SIGHAN joint Conference on Chinese Language Processing, 2012*

Liste des auteurs

Liste des mots clés