Actes de la conférence Traitement Automatique de la Langue Naturelle, TALN 2018

Anne-Laure Ligozat, Peggy Cellier, Anne-Lyse Minard, Vincent Claveau, Cyril Grouin, Patrick Paroubek

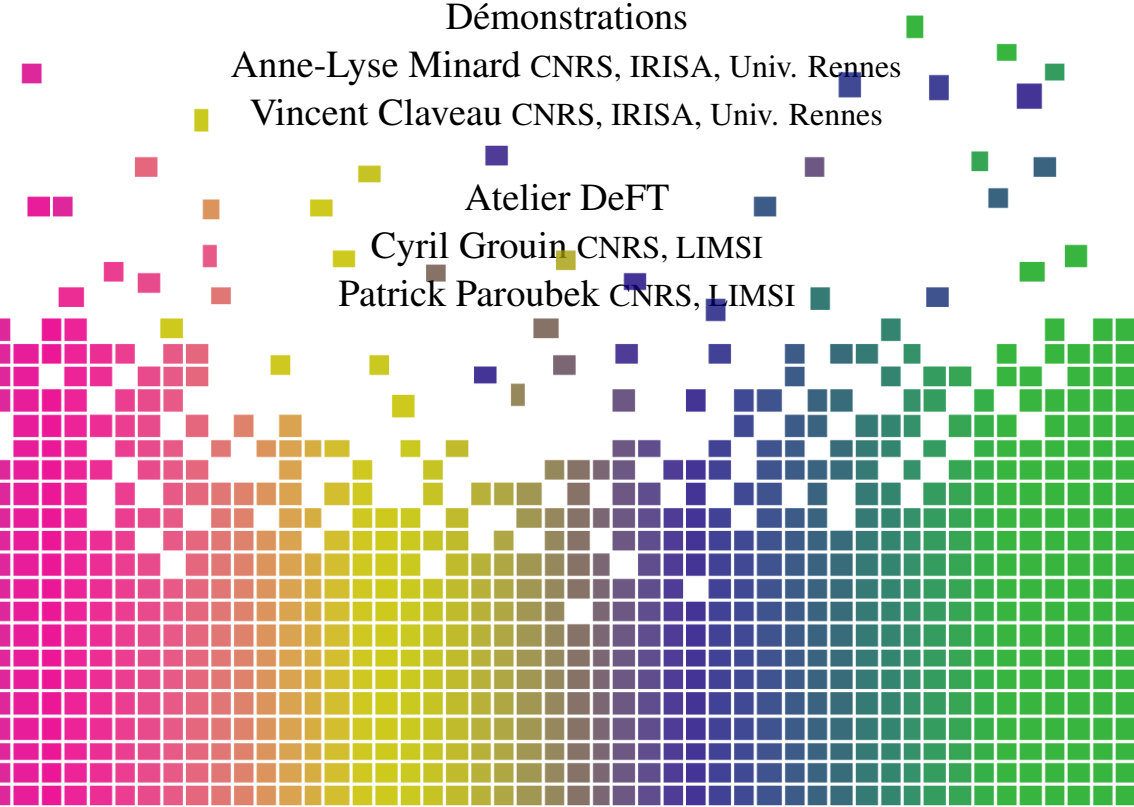
Actes de la conférence TALN 2018

Volume 2 : Démonstrations, articles des Rencontres Jeunes Chercheurs, ateliers DeFT

RJC
Anne-Laure Ligozat LIMSI, ENSIIE
Peggy Cellier IRISA, INSA Rennes

Démonstrations
Anne-Lyse Minard CNRS, IRISA, Univ. Rennes
Vincent Claveau CNRS, IRISA, Univ. Rennes

Atelier DeFT
Cyril Grouin CNRS, LIMSI
Patrick Paroubek CNRS, LIMSI



Préface

Comité d'organisation de CORIA-TALN-RJC

Coordinateur :

Vincent Claveau, CNRS, IRISA, Univ. Rennes

Webmestres :

Clément Dalloux, CNRS, IRISA, Univ. Rennes
Cédric Maigrot, IRISA, Univ Rennes

Resp. démonstrations :

Anne-Lyse Minard, CNRS, IRISA, Univ. Rennes

Resp. ateliers :

Annie Forêt, IRISA, Univ. Rennes

Resp. salon de l'innovation :

Géraldine Damnati, Orange, Lannion
Aleksandra Gerraz, Orange, Lannion

Resp. sponsoring :

Gwénolé Lecorvé, IRISA, ENSSAT, Univ. Rennes

Infographiste :

Agnès Cottais, IRISA, Rennes

Support administratif :

Élisabeth Lebret, Inria, Rennes
Aurélie Patier, IRISA, Rennes

Membres du comité d'organisation :

Cheikh Brahim El Vaigh, Inria, Rennes
Peggy Cellier, IRISA, INSA Rennes
Guillaume Gravier, IRISA, CNRS, Rennes
Pierre-François Marteau, IRISA, Univ. Bretagne Sud, Vannes
Nicolas Béchet, IRISA, IUT de Vannes
Pascale Sébillot, IRISA, INSA Rennes
Mikaïl Demirdelen, IRISA, INSA Rennes

Ainsi que les équipes techniques et administratives du centre Inria Rennes Bretagne Atlantique.

Comité de programme pour les démonstrations

- Anne-Lyse Minard, CNRS, IRISA, Univ. Rennes
- Vincent Claveau, CNRS, IRISA, Univ. Rennes

Comité de programme RJC

Présidentes du comité de programme :

- Peggy Cellier, IRISA, INSA Rennes
- Anne-Laure Ligozat, LIMSI, ENSIIE

Comité de programme :

Pegah Alizadeh, GREYC, Université Caen
Ismail Badache, LIS, Université Aix-Marseille
Nicolas Béchet, IRISA, IUT Vannes
Fatma Chamekh, Centre de recherche Léonard de Vinci Pôle Universitaire
Thierry Charnois, LIPN, Université Paris 13
Caio Corro, LIPN, Université Paris 13
Clément Dalloux, IRISA, CNRS
Antoine Doucet, L3i, Université La Rochelle
Annie Foret, IRISA, Univ. Rennes
Ophélie Fraisier, IRIT, Université Paul Sabatier
Thomas François, Université catholique de Louvain
Gaël Guilbon, LIS
Léa Laporte, LIRIS, INSA Lyon
Gwénolé Lecorvé, IRISA, ENSSAT, Université de Rennes
Vincent Letard, LIMSI
Cédric Maigrot, IRISA, Université de Rennes
Yann Mathet, GREYC, Université Caen
Anne-Lyse Minard, IRISA, CNRS
Jose Moreno, IRIT, Université Paul Sabatier
Gia-Hung Nguyen, IRIT, Université Paul Sabatier
Diana Nurbakova, LIRIS, INSA Lyon
Yannick Parmentier, LORIA, Université Lorraine
Karen Pinel-Sauvagnat, IRIT, Université Paul Sabatier
Camille Pradel, Synapse Développement
Arnaud Soulet, LI, Université François Rabelais Tours
Isabelle Tellier, Lattice, Université Paris 3
Thibaut Thonet, IRIT, Université Paul Sabatier
Yannick Toussaint, LORIA, Mines Nancy
Julien Velcin, ERIC, Université Lyon 2
Eloi Zablocki, LIP6, Sorbonne Université
Haifa Zargayouna, LIPN, Université Paris 13

Comité de programme DeFT

Comité d'organisation :

Iris Eshkol, PHILLIA - U. Paris-Nanterre
Patrick Paroubek, LIMSI, CNRS, Université Paris-Saclay
Amel Fraisse, GERIICO - U. Lille3
Vincent Claveau, CNRS, IRISA
Cyril Grouin, LIMSI, CNRS, Université Paris-Saclay
Thierry Hamon, LIMSI, CNRS, Université Paris-Saclay, Université Paris XIII

Comité scientifique :

Patrice Bellot, LSIS
Farah Benamara, IRIT
Vincent Claveau, CNRS, IRISA
Iris Eshkol, PHILLIA - U. Paris-Nanterre
Amel Fraisse, GERIICO
Cyril Grouin, LIMSI-CNRS
Vincent Guigue, LIP6
Thierry Hamon, LIMSI-CNRS
Agata Jackiewicz, Praxiling, Université Montpellier 3
Jihen Karoui, LIUM
Laura Monceau, LINA
Véronique Moriceau, IRIT
Viviana Patti, U. Torino
Mathieu Roche, CIRAD
Juan-Manuel Torres-Moreno, LIA

TABLE DES MATIÈRES

Articles RJC

Démonstrations

Articles DeFT

Articles RJC

Construction de patrons lexico-syntaxiques d'extraction pour l'acquisition de connaissances à partir du web

Chloé Monnin[1] Olivier Hamon[1,]
(1) Syllabs, 35-37 rue Chanzy, 75011 Paris, France
monnin@syllabs.com, hamon@syllabs.com

RESUME

Cet article présente une méthode permettant de collecter sur le web des informations complémentaires à une information prédéfinie, afin de remplir une base de connaissances. Notre méthode utilise des patrons lexico-syntaxiques, servant à la fois de requêtes de recherche et de patrons d'extraction permettant l'analyse de documents non structurés. Pour ce faire, il nous a fallu définir au préalable les critères pertinents issus des analyses dans l'objectif de faciliter la découverte de nouvelles valeurs.

ABSTRACT

Relation pattern extraction and information extraction from the web.
This article presents an information extraction method which collects additional information on the web so as to enrich already existing information and then fill in a knowledge base. Our method is based on lexical and syntactical patterns, both used as search queries and extraction patterns to allow the analysis of unstructured documents. To do so, we first defined relevant criteria coming from the analysis phase so as to ease the discovery of new values.

MOTS-CLES : Construction de patrons, extraction d'information, extraction d'entités nommées, syntaxe en dépendances, apprentissage de patrons d'extraction, web comme corpus.

KEYWORDS: Pattern construction, information extraction, named entities recognition, dependancy syntax, extraction pattern learning, web as a corpus.

1 Introduction

L'utilisation du web comme corpus d'analyse n'est pas une nouveauté. En effet, avec l'essor des moteurs de recherche dans la seconde moitié des années 90, l'exploration des contenus analysables linguistiquement était facilitée. Cependant, malgré le formatage requis pour l'interprétation des pages web par le navigateur ou l'apparition de standards tels que *schema.org*[1], les textes accessibles *via* un moteur de recherche restent du texte 'libre', contenant des informations, mais qui ne peuvent être interprétées directement par la machine autrement que comme des chaînes de caractères. L'analyse de ces textes devient par conséquent une nécessité pour en extraire de l'information en contexte, notamment pour la mettre en relation avec d'autres informations.

Nous proposons ici une méthode permettant de construire à partir du web des patrons lexico-syntaxiques d'extraction, ainsi que leur application pour extraire des informations similaires et/ou complémentaires à une information donnée.

[1] http://schema.org

Nos travaux se placent dans un contexte d'enrichissement de base de connaissances à Syllabs, dans le domaine de la génération automatique de textes où le besoin de nouvelles données est indispensable et se fait croissant. Si des bases de données existent pour de nombreuses thématiques, elles sont bien souvent incomplètes, voire erronées. De plus, la mise à disposition des bases prend du temps, impactant l'accès à ces données. Ainsi, il nous paraît nécessaire d'aller chercher de l'information là où elle se trouve, et en particulier sur le web. La masse de documents accessibles *via* les moteurs de recherche laisse penser que, dans leur grande majorité, les données les plus fréquentes seront correctes et, quand bien même le doute subsisterait, une validation semi-manuelle est toujours envisageable. Mais avant toute chose, les données doivent être extraites.

Le choix de travailler à l'aide de patrons d'extraction est motivé par l'aspect linguistique de nos travaux. En effet, les patrons résultant de notre implémentation ne sont pas seulement un ensemble de traits permettant l'extraction d'informations, mais aussi des représentations linguistiques de la formulation d'une relation entre deux valeurs (ou plus), apportant de la connaissance supplémentaire et qui pourrait permettre d'étudier la variabilité linguistique par exemple. Une autre application de ces patrons comme outil d'extraction d'entités nommées peut également être d'extraire de nouvelles entités qui ne sont pas connues actuellement.

Nous allons dans un premier temps revenir sur différentes méthodes d'extraction présentes dans la littérature, puis présenter notre méthode. Nous fournirons ensuite nos résultats sur une extraction à partir d'une paire de valeurs, ainsi que des résultats d'évaluation. Nous conclurons sur les optimisations et extensions envisagées pour l'amélioration de la méthode.

2 Etat de l'art

L'extraction d'informations a toujours été un domaine prédominant dans le traitement automatique du langage. Cette tâche consiste en la détection et l'extraction sous forme structurée, de données présentes dans des documents non structurés. Le développement des méthodes d'extraction a été stimulé par les conférences MUC (*Message Understanding conferences*) dès la fin des années 80, en organisant des compétitions financées par la DARPA (Grishman & Sundheim, 1996). L'objectif de ces conférences était d'extraire le plus d'informations possible sur des thèmes bien déterminés et d'évaluer les systèmes d'extraction d'informations selon une grille d'évaluation commune. En dehors de ces conférences, la communauté scientifique a développé d'autres méthodes, utilisant notamment l'extraction de relations à l'aide de patrons.

Brin (1998) avec son système DIPRE (*Dual Iterative Pattern Relation Extraction*) a proposé une méthode permettant d'exploiter le potentiel de la multiplicité de sources d'informations présentes sur le web. Cette méthode génère des patrons à partir d'exemples de valeurs fournies en entrée par l'utilisateur. Les patrons extraits sont ensuite utilisés pour rechercher de nouvelles instances de la relation que le patron représente. Agichtein & Gravano (2000) approfondissent le travail de Brin avec Snowball. Leurs contributions apportent entre autres une proposition d'évaluation du système ainsi qu'une méthode pour mesurer le poids des patrons extraits. Cependant, leur méthode est adaptée à des collections de documents hors web.

Les méthodes présentées ci-dessus sont supervisées, dans le sens où les outils reçoivent en entrée des valeurs définies préalablement. Que ce soit simplement un couple de valeurs pour DIPRE ou une représentation formelle d'une relation, cette entrée est le point de départ de la méthode.

Les méthodes suivantes sont semi-supervisées, puisque leur point d'entrée n'est pas une valeur mais un corpus. En effet, contrairement aux méthodes ci-dessus, celles-ci ont pour but d'extraire toutes les relations au sein d'un corpus défini, et non une relation particulière servant d'amorce.

Lin & Pantel (2001) font l'hypothèse distributionnelle pour développer DIRT, dont le but est d'extraire des relations d'inférence (dans ce cas précis des paraphrases). Leur méthode permet d'extraire ce qu'ils nomment des *slot fillers*, soit des entités qui apparaissent dans les trous laissés par leur patrons à partir d'arbres syntaxiques en dépendances, de règles de comparaison et d'un calcul de similarité afin d'identifier les paraphrases. Etzioni et al. (2004) publient les premiers résultats de leur système KnowItAll, un outil permettant d'extraire ce qu'ils nomment des faits, soit des informations mises en relations. Leurs travaux se poursuivront les années suivantes avec KnowItNow (Cafarella et al., 2005). Leur méthode s'appuie sur des clauses, soient des règles prédéfinies pour extraire des entités qui répondent à ces règles. Les améliorations de KnowItNow concernent principalement la rapidité d'exécution de l'outil ainsi que l'apprentissage de règles d'extraction. Etzioni et al. (2011) approfondissent leur système et développent Reverb, un système d'extraction de relations basées sur les verbes. Ce système applique un filtre syntaxique reposant sur des patrons morphosyntaxiques (des expressions régulières de parties du discours) et un filtre lexical. Plus récemment, Akbik et al. (2012) ont étendu l'extraction de relation non supervisée en extrayant toutes les relations d'un corpus puis en les classant par clustering.

Les travaux cités ci-dessus sont implémentés pour des contenus en anglais. Pour un contexte français, il est possible de trouver des méthodes similaires appliquées à des documents techniques, comme par exemple le système Prométhée (Morin, 1999) qui extrait des schémas lexico-syntaxiques représentatifs d'une relation sémantique.

Les méthodes d'extraction de relations et d'informations à partir de patrons sont très développées au sein des méthodes d'extraction en général. Dans la conception de notre outil, nous nous sommes beaucoup inspirés des travaux que nous évoquons ci-dessus. Notre méthode utilise aussi bien des analyses syntaxiques en dépendances (Akbik) que des patrons d'extraction (Etzioni, Agichtein, Morin). De plus notre corpus d'extraction est également le web, comme (Brin).

3 Présentation de la méthode

Nous décrivons dans un premier temps la méthode de construction de patrons d'extractions ainsi que les outils utilisés, en prenant l'exemple des valeurs d'entrée suivantes : "*Victor Hugo*" et "*26 février 1802*", soit un nom et une date de naissance. Victor Hugo étant un personnage dont la vie a bien été documentée, nous avons estimé qu'il ferait un bon point de départ pour extraire différentes façons de renseigner une date de naissance. Puis, nous décrivons la méthode d'extraction de nouvelles valeurs à partir de ces patrons. La Figure 1 ci-dessous résume l'ensemble de la méthode.

3.1 Construction des patrons

La méthode de construction des patrons d'extraction se déroule selon plusieurs étapes. Tout d'abord, nous utilisons les valeurs d'entrée comme requête de moteur de recherche afin de recueillir des contenus textuels. Ces textes sont ensuite segmentés en phrases, qui sont filtrées selon la pertinence de l'information présente. Les phrases sélectionnées sont ensuite analysées syntaxiquement et les patrons sont construits à partir de ces analyses. Nous détaillons chacune de ces étapes ci-dessous.

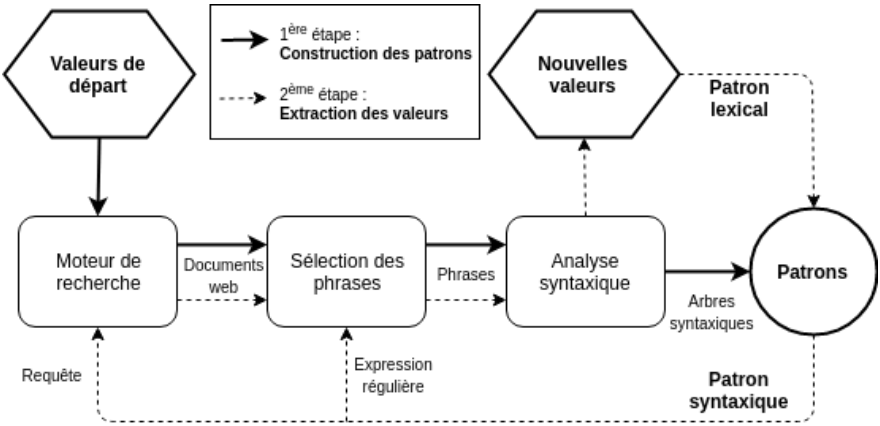


FIGURE 1: Schéma de notre méthode

3.1.1 Collecte de contenus sur le Web

La première étape consiste à rassembler des documents hétérogènes à partir desquels nous construirons les patrons. Les documents sont collectés sur le web : des valeurs passées en entrée ("Victor Hugo" et "2 février 1802") sont transformées en requêtes adaptées au moteur de recherche. La requête est ensuite envoyée à un moteur de recherche qui retourne des URLs (100 dans le cadre de nos tests). Pour extraire le texte utile des pages web, nous utilisons un extracteur appliquant l'algorithme BTE (Fin et al., 2001). Au terme de cette extraction, nous obtenons une série de contenus textuels et leurs URLs associées.

3.1.2 Sélection des phrases

Le texte que nous avons collecté lors de la précédente étape est segmenté en phrases, qui sont ensuite triées. Pour la segmentation nous utilisons le tokenizer PunktTokenizer (Kiss & Strunk, 2006) tel qu'il est implémenté dans le package NLTK. Nous sélectionnons ensuite les phrases d'après deux critères : la présence exacte de toutes les valeurs recherchées et la longueur de la phrase qui doit faire moins de 200 tokens pour être analysée.

Ce dernier critère est lié à la quantité d'informations que l'analyseur syntaxique peut traiter. Au-delà de 200 tokens, les relations à analyser requièrent trop de mémoire. Par ailleurs, après observation du corpus, ces séquences ne sont souvent pas pertinentes par rapport à ce que nous souhaitons extraire. La segmentation en phrases ne s'appliquant pas par exemple aux listes ou aux contenus de tableaux, ces séquences sont souvent de longues énumérations qui ne contiennent pas d'informations exprimant une relation explicite.

3.1.3 Analyse syntaxique des phrases sélectionnées

L'analyse syntaxique des phrases sélectionnées lors de la phase précédente est ensuite réalisée à l'aide de Talismane (Urieli, 2013), un analyseur syntaxique en dépendances. L'arbre syntaxique correspondant à une phrase est une liste ordonnée de nœuds possédant chacun des attributs tels que :

- l'identifiant du nœud correspondant à sa position dans la phrase,
- le token : la forme lexicale d'un mot dans la phrase,
- le lemme : la forme non fléchie du token,
- la partie du discours,
- les traits morphosyntaxiques du token : genre, nombre, personne, etc.,
- l'identifiant du token gouverneur,
- le lien de dépendance entre le token et son gouverneur.

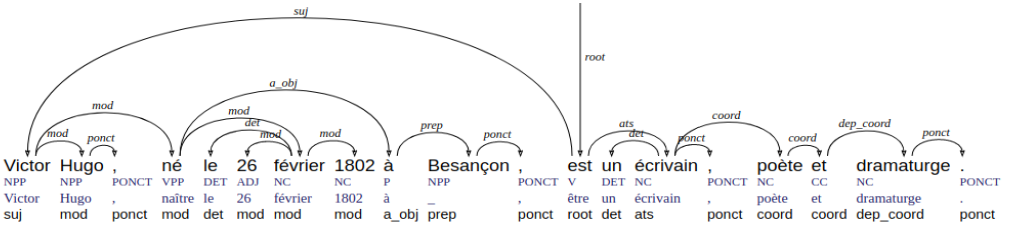


FIGURE 2 : Exemple d'arbre syntaxique[2]

Pour créer les patrons nous utilisons plusieurs de ces traits, à savoir les parties du discours, les liens de dépendances et la forme du token. Les liens de dépendances entre les valeurs présentes dans la phrase vont nous permettre d'établir une relation entre ces valeurs, tandis que les parties du discours et les formes lexicales nous serviront à la construction du patron en tant que tel.

3.1.4 Sélection des relations

Les relations entre les valeurs des arbres syntaxiques obtenus précédemment sont sélectionnées afin de créer les patrons. Nous considérons qu'une relation est présente dans une phrase s'il existe un chemin reliant les valeurs de départ. Chaque arbre est analysé comme un graphe non orienté[3], où chaque nœud correspond à un token de la phrase. Nous parcourons les nœuds et déterminons le chemin le plus court entre les valeurs de départ.

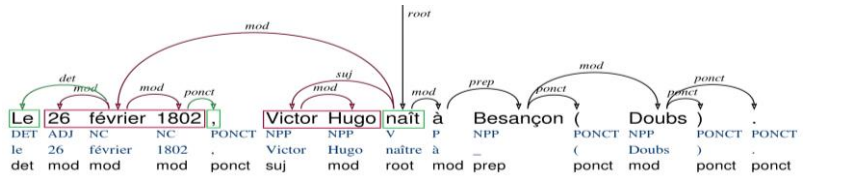


FIGURE 3 : Extraction du chemin
En rouge le chemin le plus court, en vert les tokens récupérés sur le chemin.

Une fois ce chemin déterminé, nous conservons les nœuds correspondant aux valeurs ainsi que ceux présents sur le chemin. Les chemins extraits correspondent aux sous arbres explicitant la relation entre les valeurs de départ, soit un ensemble de nœuds avec leurs traits associés. Une sélection des arbres s'opère ici puisque tous ne sont pas pertinents. En effet, dans certains cas il n'y a pas de chemin entre les valeurs parce qu'il n'y a pas de relation explicite dans la phrase (par ex. : "*26 février 1802 : Victor Hugo…*").

[2] La visualisation des arbres syntaxiques a été réalisée *via* Arborator https://arborator.ilpga.fr

[3] L'orientation dans les arbres syntaxiques sert à formaliser la relation hiérarchique entre les différents mots de la phrase. Seuls les liens sont utilisés ici, en faisant abstraction de la hiérarchie.

3.1.5 Définition des patrons lexico-syntaxique

Les patrons d'extraction sont enfin créés à partir des sous arbres sélectionnés et reliant les valeurs. Nous définissons un patron comme un objet composé de trois éléments :

1. Une représentation lexicale (sous phrase anonymisée),
2. Un ensemble de traits syntaxiques (patron syntaxique),
3. Une liste de catégories d'entités nommées attendues par le patron.

La création du patron se fait en deux étapes. La première, l'anonymisation, a pour objectif d'identifier les entités nommées dans la sous-phrase et de construire la partie lexicale du patron. La seconde est la sélection des attributs du sous arbre qui permettront de composer le patron syntaxique.

3.1.6 Construction du patron lexical

Nous analysons un tronçon à anonymiser à l'aide de l'extracteur d'entités nommées construit à base de règles afin de remplacer les entités trouvées par les caractères ".*", ce qui permet la réutilisation du patron pour l'extraction de nouvelles valeurs. Cette chaîne anonymisée correspond à la représentation lexicale du patron tandis que la liste des types d'entités nommées détectées est également conservée comme attribut du patron lexico-syntaxique. Pour l'extraction des entités nommées nous utilisons un système développé en interne à Syllabs (Ma et al., 2011). Les différents types d'entités concernent les noms de personne (*Person*), les noms de lieu (*Geo*), les dates (*Date*), les noms de profession ou de fonction (*Function*) et les noms d'organisation (*Organisation*).

Les sous-phrases sont construites à partir de tous les tokens sur le chemin reliant les valeurs. Parmi ces tokens, d'éventuelles entités nommées qui n'étaient pas présentes dans les valeurs de départ sont extraites. Par exemple dans la sous phrase "*Victor Hugo est né à Besançon le 26 février 1802*", nous identifions avec l'extraction d'entités nommées une nouvelle valeur, à savoir *Besançon*, le lieu de naissance de *Victor Hugo*. Le patron construit permettra d'extraire 3 valeurs : un nom, une date de naissance et un lieu de naissance.

3.1.7 Construction du patron syntaxique

La construction du patron syntaxique se fait à partir de certains attributs obtenus lors de l'analyse syntaxique. Ces attributs sont obtenus à partir des entités anonymisées lors de l'étape précédente. Pour chaque entité, les traits retenus dans le patron syntaxique sont la partie du discours, la forme lexicale du token du nœud gouverneur, et les éventuels enfants : si la partie du discours d'un nœud donné est un nom commun, un nom propre, une préposition simple ou composée, on ne pourra pas déterminer, lors de l'extraction, ni le nombre ni la nature des nœuds gouvernés par le nœud donné, ce trait est donc catégorisé en optionnel ('*optional*'). Pour toute autre partie du discours, le lien de dépendance et la position des enfants dans la phrase sont retenus pour ce trait.

```
{0: {u'children': 'optional',
     'gov': {u'né': u'suj'},
     'tag': u'NPP'},
1: {u'children': 'optional',
    'gov': {u'né': u'mod'},
    'tag': u'NC'}}
```

FIGURE 4 : Exemple de patron syntaxique correspondant au patron lexical ".*, *né le* .*"
Chaque entrée du patron correspond à une entité à extraire et ses attributs attendus

3.2 Utilisation des patrons lexico-syntaxiques pour extraire de nouvelles valeurs

Dans la section précédente nous avons présenté notre méthode de construction de patrons lexico-syntaxiques du web à partir de valeurs prédéfinies. Ces patrons vont nous permettre d'extraire de nouvelles valeurs comparables à celles de départ. Les deux méthodes sont, au final, assez proches dans leur fonctionnement. Nous utilisons les patrons lexicaux comme requêtes pour le moteur de recherche afin d'extraire des documents web et comme expressions régulières pour sélectionner les phrases pertinentes (1). Ces phrases sont analysées syntaxiquement et le patron syntaxique est utilisé pour en extraire de nouvelles valeurs (2). Ces valeurs sont enfin validées au cours de la phase d'identification, à l'aide d'un extracteur d'entités nommées (3).

3.2.1 Utilisation des patrons lexicaux pour la sélection de phrases

Afin de rassembler des documents pertinents pour l'extraction de nouvelles valeurs, nous utilisons dans un premier temps les représentations lexicales des patrons comme requêtes passées au moteur de recherche. En effet, la présence de caractères ".*" à la place des valeurs est considérée comme un joker par les moteurs de recherche[4]. Ainsi, un patron représenté par *".* est né le .* à .*"* permettra de retrouver des documents contenant des phrases correspondant au patron. Par ailleurs les caractères ".*" sont également des jokers dans la syntaxe des expressions régulières, que nous utilisons dans l'étape suivante. Comme dans la section précédente, chaque requête renvoie jusqu'à 100 URLs. Nous utilisons également le même algorithme d'extraction du texte utile des documents web retournés. Le patron lexical est appliqué aux contenus collectés en tant qu'expression régulière. Les segments détectés par le patron sont conservés puis analysés.

3.2.2 Extraction de nouvelles valeurs à l'aide des patrons syntaxiques

Les segments sélectionnés au cours de l'étape précédente sont analysés syntaxiquement avec Talismane afin d'obtenir les mêmes traits syntaxiques que ceux de la section précédente. Les patrons syntaxiques sont appliqués aux arbres issus de l'analyse. Nœud après nœud, les traits du patron sont comparés de manière itérative à ceux de l'arbre en commençant par la partie du discours, suivie du nœud gouverneur. Si un nœud de l'arbre possède les mêmes traits qu'un nœud du patron, le nœud et ses éventuels enfants sont conservés. Par exemple le patron suivant *".* est né à .* le .*"*, attend pour la première valeur à extraire les traits `{pos : NPP, gov : (né, suj)}`. Ainsi, la phrase "*Thomas est né à Rouen (France), le 27 février 1978.*" (dont l'analyse syntaxique est présentée en Figure 5) possède bien un nom propre (NPP) sujet du verbe *naître* (suj). La première valeur est ainsi extraite.

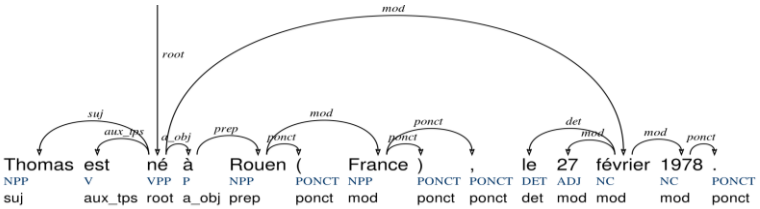


FIGURE 5 : Exemple de phrase sélectionnée après analyse syntaxique

[4] Exemple de syntaxe Google : https://support.google.com/websearch/answer/2466433

Nous procédons ainsi jusqu'à ce que toutes les valeurs du patron soient trouvées. Si un segment ne contient pas le nombre de valeurs attendu par le patron, elle n'est pas sélectionnée. Au terme de cette extraction nous conservons l'ensemble des segments sélectionnés et leurs valeurs extraites.

```
    "values": { "<Date>": "27 février 1978",
                "<Geo>": "Rouen France",
                "<Person>": "Thomas" },
    "sentence": "Thomas est né à Rouen ( France ) , le 27 février 1978 ."
```

FIGURE 6 : Exemple de valeurs extraites

3.2.3 Identification des entités nommées

Les entités nommées sont ensuite analysées à partir des segments sélectionnés lors de l'extraction des valeurs afin de valider que les valeurs extraites du segment correspondent bien au type d'entités attendues par le patron. L'extracteur d'entités nommées de Syllabs permet le typage des entités.

```
"values": { "<Date>": "27 février 1978",
            "<Geo>": "Rouen France",
            "<Person>": "Thomas" },
"sentence": "Thomas est né à Rouen ( France ) , le 27 février 1978 .",
"entities": [ {'name': 'Geo', 'text': u'France'},
              {'name': 'Person', 'text': u'Thomas'},
              {'name': 'Geo', 'text': u'Rouen'},
              {'name': 'Date', 'text': u'27 février 1978'}]
```

FIGURE 7 : Exemple de résultats de l'identification des entités nommées

Comme nous pouvons le voir dans la Figure 7, le type des valeurs extraites par le patron correspond ici à celui attendu. Dans le cas où nous aurions plusieurs valeurs possibles parmi les résultats (par ex. un nom et deux dates de naissances possibles), nous prenons celle qui a le plus d'occurrences.

4 Exemple d'extraction simple

À l'aide des patrons lexico-syntaxiques construits lors de la première phase de notre méthode, nous avons réussi à extraire de nouvelles valeurs similaires et complémentaires à celles de départ. Avec les deux valeurs présentées en exemple en entrée, nous avons extrait 529 ensembles de valeurs (uniques) dont 322 qui contenaient de nouvelles entités (c.-à-d. lieu de naissance et/ou profession). Nous détaillons dans cette section les résultats des deux phases de constructions des patrons et d'extraction de nouvelles valeurs, à partir de l'exemple *"Victor Hugo+26 février 1802"*.

4.1 Construction de patrons

Au terme de la première phase de collecte de documents sur le web, nous avons obtenu 91 documents (certaines des 100 réponses étant des liens morts). Après la phase de sélection des phrases contenant les valeurs, 47 d'entre elles ont été conservées, puis 47 arbres syntaxiques suite à l'analyse syntaxique, desquels les relations entre les valeurs de départ sont extraites. Une fois la sélection des relations terminée, nous avons obtenu 34 sous arbres syntaxiques soit, au terme de la phase de création des patrons, 34 patrons lexico syntaxiques permettant l'extraction de noms de personnes et leur date de naissance et, dans la majorité des cas (26 patrons), de nouvelles informations, à savoir le lieu de naissance et la profession.

4.2 Extraction de nouvelles valeurs

Les 34 patrons extraits sont appelés comme requêtes dans un moteur de recherche, nous donnant 3 400 documents, soit près de 25 000 segments à analyser. Nous détaillons les résultats pour un patron particulier *".* est né à .* le .*"* qui nous a permis d'extraire 19 ensembles de valeurs après analyse syntaxique et identification des entités nommées. La Figure 8 présente quelques valeurs extraites par ce patron. Au total, avec les 34 patrons nous avons extrait 1 380 valeurs. Parmi les 1 380 valeurs extraites, 653 sont validées par l'identification des entités nommées, soit 47 %.


```
"values": { "Person": "Jacques De Decker",
            "Geo": "Bruxelles",
            "Date": "le 19 août 1945" },
"sentence": "Jacques De Decker est né à Bruxelles le 19 août 1945 ."

"values": { "Person": "Yann Arthus-Bertrand",
            "Geo": "Paris",
            "Date": "le 13 mars 1946" },
"sentence": "Yann Arthus-Bertrand est né à Paris le 13 mars 1946 ."

"values": { "Person": "Thomas",
            "Geo": "Rouen France",
            "Date": "le 27 février 1978" },
"sentence": "Thomas est né à Rouen ( France ) , le 27 février 1978 ."
```


FIGURE 8 : exemples de valeurs trouvées par le patron ".* *est né à* .* *le* .*"

D'autres exemples de patrons trouvés sont présentés dans le Tableau 1.

né le <Date> à <Geo> dans le <Geo> , <Person>
<Person> est né le <Date>
<Person> est né à <Geo> le <Date> ,
<Person> naît le <Date> à
la naissance de <Person> : <Person> est né le <Date>
Troisième et dernier enfant de <Person> et <Person> , <Person> naît à <Geo> le <Date> .
Éphéméride <Date> naissance de <Person> <Date>_11:05 Classé

TABLE 1 : Exemples de patrons construits à partir des valeurs "Victor Hugo" et "*26 février 1802*"

5 Evaluation de la méthode

Dans cette section, nous mesurons la capacité de notre méthode à extraire des informations pertinentes à partir de la requête "*Victor Hugo+26 février 1802*", à savoir des dates de naissance de personnalités sélectionnées dans un jeu d'évaluation issu de la base de connaissances DBpedia[5]. Pour ce faire, nous avons sélectionné 100 noms de personnalités françaises au hasard ainsi que leurs dates de naissance servant, elles, de référentiel.

[5] http://wiki.dbpedia.org

5.1 Protocole

Les noms des personnalités sont utilisées comme amorce et sont susceptibles, en les associant aux patrons d'extraction, de permettre l'identification des dates de naissance à partir du web. Pour chaque patron, la valeur correspondant à l'entité de type nom est donc remplacée par le nom d'une personnalité. L'objectif est ici l'enrichissement d'une base de connaissances.

Les patrons d'extractions augmentés des noms de personnalités sont passés comme requête au moteur de recherche. Les documents retournés sont collectés et les segments pertinents sont sélectionnés en appliquant les patrons lexicaux comme expression régulière (cf. section 2.1). Une fois les segments sélectionnés, ils sont analysés syntaxiquement et les patrons syntaxiques sont appliqués (cf. section 1.3). Enfin, les valeurs de type *Date* sont extraites, normalisées (c.-à-d. deux chiffres pour le jour, le mois en toutes lettres et les quatre chiffres de l'année), puis la date la plus fréquemment trouvée pour chaque nom est comparée à celle de DBpedia.

La précision (p), le rappel (r) et la f-mesure (f1) ont été utilisées et sont définies selon :

$$p = nb\ de\ valeurs\ correctes\ trouvées \div nb\ de\ valeurs\ trouvées$$
$$r = nb\ de\ valeurs\ correctes\ trouvées \div nb\ de\ valeurs\ à\ trouver$$
$$f1 = 2 * (p * r) / (p + r)$$

5.2 Résultats

5.2.1 Précision des patrons

Au terme de l'extraction, nous trouvons des valeurs pour 10 des 34 patrons d'extraction. Parmi ces 10 patrons, 4 d'entre eux n'ont trouvé aucune valeur correcte, soit une précision de 60 %. La précision pour l'ensemble des patrons est de 17 %. Le Tableau 2 présente le nombre des valeurs trouvées et correctes pour chacun des patrons présentés dans le Tableau 1.

Patron	Valeurs trouvées	Valeurs correctes
né le .* à .* dans le .* , .*	0	0
.* est né le .*	48	34
.* est né à .* le .* ,	1	0
.* naît le .* à	4	2
la naissance de .* : .* est né le .*	0	0
Troisième et dernier enfant de .* et .* , .* naît à .* le .* .	0	0
Éphéméride .* naissance de .* .*_11:05 Classé	0	0

TABLE 2 : Quelques exemples de patrons et le nombre de valeurs extraites par chacun d'entre eux

5.2.2 Evaluation de la méthode selon les valeurs identifiées

Concernant les scores d'évaluation par valeur, sur les 100 dates de naissance à trouver initialement, des valeurs ont été extraites pour 46 d'entre elles. Parmi ces 46 valeurs, et après normalisation, 34 correspondent aux dates indiquées dans DBpedia. La précision moyenne pour les 46 valeurs brutes extraites est de 74 % et le rappel sur l'ensemble des valeurs est de 34 %, ce qui donne une f-mesure de 47 %. Nous présentons ici deux résultats qui ne sont pas corrects et en expliquons les causes.

Le premier exemple est présenté en Figure 9. Dans cet exemple nous constatons que la date extraite n'est pas fausse, mais incomplète. En effet, nous recherchons des dates complètes (jour, mois et année), tandis qu'ici seule la date est spécifiée dans la phrase.

```
{ ".* est né le .*": [
    { "values": { "<Date>": "14 juillet",
                  "<Person>": "Marine ce joli bébé de 2,080 kg" },
      "sent": "Marine , ce joli bébé de 2,080 kg , est né le 14 juillet à 13 h
25 à la maternité de Chalon-sur-Saône ." } ] }
```

FIGURE 9: un exemple de résultat incomplet

Le second exemple, présenté en Figure 10, est un cas de mauvais découpage en phrases. En effet, l'expression régulière de ce patron extrayant tous les termes avant "Naissance", et la phrase étant mal découpée (probablement dû à l'absence de ponctuation entre les deux propositions "*Encyclopedia Universalis Repères biographiques*" et "*16 septembre 1867 Naissance de Jean-Baptiste Charcot à Neuilly-sur-Seine* ."), l'analyse syntaxique est faussée et tous les termes avant la date sont extraits.

```
{ ".* Naissance de .* à": [
    { "values": { "<Date>": "Encyclopedia Universalis Repères biographiques 16
septembre 1867",
                  "<Person>": "Jean-Baptiste Charcot" },
      "sent": "Source : Encyclopedia Universalis Repères biographiques 16
septembre 1867 Naissance de Jean-Baptiste Charcot à Neuilly-sur-Seine ." } ] }
```

FIGURE 10 : un exemple de résultat mal extrait

5.3 Conclusions de l'évaluation

Après observation des résultats, nous constatons que parmi les 34 patrons d'extraction initiaux, 24 d'entre eux ne trouvent aucune valeur, parce qu'ils contiennent trop de bruit (nous proposons un début de réflexion dans la section suivante concernant l'amélioration de la sélection de patrons.). Toutefois, jusqu'à présent, nous avons conservé l'ensemble des patrons afin d'être le plus exhaustif possible, obtenir un grand nombre de valeurs, et ainsi se fier à la valeur ayant le plus grand nombre d'occurrence. Notre méthode nous permet d'atteindre une précision de 74% pour l'extraction de dates seules, sachant que les erreurs viennent parfois de résultats incomplets. Cependant, nous devons revoir certaines phases de la méthode, notamment le découpage en phrases afin d'améliorer l'analyse syntaxique et par conséquent l'extraction de nouvelles valeurs.

6 Améliorations

Nous présentons dans cette section nos deux principales pistes d'amélioration de la méthode pour l'enrichissement de connaissances à partir du web.

6.1 Sélection des patrons avant la recherche de valeurs

Comme nous avons pu l'observer au cours de l'évaluation DBpedia, certains patrons ne permettent pas du tout d'extraire des valeurs. De plus, parmi les patrons construits, beaucoup sont très proches, avec parfois seulement un caractère de différence. Ainsi, nous avons cherché à sélectionner les patrons "utiles" avant l'extraction de nouvelles valeurs. La première question a été de déterminer quels seraient les critères de sélection d'un patron d'extraction. En effet, d'une extraction à l'autre, les patrons sont très différents, et notre méthode doit s'adapter à tous les cas de figure.

Nous avons envisagé d'utiliser la mesure de l'entropie de Shannon (2001) pour classer les patrons selon la quantité d'information. Le choix du calcul de l'entropie est motivé par la définition même de cette mesure, qui peut être calculée simplement à partir de chaînes de caractères, indépendamment de la langue ou du contexte.

On constate que plus le patron est petit, plus son score d'entropie est faible. Dans le cas des patrons de naissances que nous avons utilisé dans cet article, nous pouvons faire un lien entre le score de précision du patron et celui de l'entropie. Il est ainsi possible d'émettre l'hypothèse que plus le patron est simple, plus il est efficace. Cependant, si cette hypothèse se vérifie, nous devrons encore déterminer un seuil de sélection du patron, soit par rapport à un score défini, soit par rapport à une proportion de patrons à sélectionner. Cette question est encore à l'étude.

6.2 Extractions à distance

Nous avons défini par extraction à distance le fait d'essayer d'appliquer l'outil d'extraction non seulement au sein d'une phrase, mais aussi à de plus grandes distances, par exemple sur l'ensemble d'un document. Notre problème se rapprochant de la résolution de chaînes de coréférences, nous avons implémenté une méthode inspirée de (Hobbs, 1986) et (Lappin & Leass, 1994) pour retrouver l'antécédent d'un pronom sujet. L'idée principale est de retrouver le nom de l'entité à laquelle le pronom fait référence en se basant sur les informations morphosyntaxiques et l'arbre syntaxique des phrases précédentes.

Ainsi, pour une paire de phrases comme "Deborah Gibson naît le 31 août 1970 à Brooklyn. Elle est la troisième d'une famille de quatre enfants."[6], notre objectif est de lier les deux phrases et d'obtenir les informations comme quoi *Deborah Gibson est née le 31 août 1970 à Brooklyn*, et *Deborah Gibson est la troisième d'une famille de quatre enfants*.

Talismane possédant un niveau d'analyse morphologique, nous avons ajouté la méthode dans la chaîne de traitement. Ainsi, lorsque nous trouvons un pronom sujet, nous gardons les informations de genre et de nombre et tentons de retrouver la première entité dans les phrases précédentes qui partagent ces informations morphologiques. Cependant, Talismane n'identifie pas assez bien les noms, et notamment leur genre, et par conséquent ne les étiquette pas tous en genre et en nombre. Pour pouvoir utiliser cette solution de résolution de chaînes de coréférences, nous pensons réentrainer Talismane sur un corpus de noms annotés en genre et en nombre.

7 Conclusion et travaux futurs

Nous avons présenté une méthode de construction semi-supervisée de patrons de relation et d'extraction en français, à partir du web. Nous avons montré que notre méthode permettait d'extraire des valeurs similaires (nom, date) et complémentaires (lieu, fonction) à partir d'un unique jeu de valeurs en entrée. Cependant, notre méthode n'est pas complète, et comporte encore des faiblesses, notamment au niveau de la sélection des patrons pour l'extraction de nouvelles valeurs. Par ailleurs, l'utilisation d'arbres syntaxiques en dépendances et la disponibilité de ressources multilingues nous permettrait de porter la méthode dans d'autres langues que le français. Une de nos prochaines étapes de réflexion sera également de porter la méthode à des relations de plus de deux valeurs.

[6] Exemple issu de la page Wikipedia : https://fr.wikipedia.org/wiki/Deborah_Gibson

Références

AGICHTEIN E., GRAVANO L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries* (pp. 85-94). ACM.

AKBIK A., VISENGERIYEVA L., HERGER P., HEMSEN H., LÖSER A. (2012). Unsupervised Discovery of Relations and Discriminative Extraction Patterns. In 24th *International Conference on Computational Linguistics*, COLING 2012 (pp. 17-32).

BRIN S. (1998). Extracting patterns and relations from the World Wide Web. In *International Workshop on the World Wide Web and Databases* (pp. 172-183). Springer, Berlin, Heidelberg.

CAFARELLA M., DOWNEY D., SODERLAND S., ETZIONI O. (2005). KnowItNow: Fast, scalable information extraction from the web. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 563-570). Association for Computational Linguistics.

ETZIONI O., CAFARELLA M., DOWNEY D., KOK S., POPESCU A., SHAKED T., SODERLAND S., WELD D., YATES A. (2004). Web-scale information extraction in knowitall. In *Proceedings of the 13th international conference on World Wide Web* (pp. 100-110). ACM.

ETZIONI O., FADER A., CHRISTENSEN J., SODERLAND S., MAUSAM M. (2011). Open Information Extraction: The Second Generation. In *IJCAI* (Vol. 11, pp. 3-10).

FINN A., KUSHMERICK N., SMYTH B. (2001). Fact or fiction: Content classification for digital libraries. *Joint DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries*. Dublin.

GRISHMAN R., SUNDHEIM B. (1996). Message Understanding Conference - 6: A Brief History. In: *Proceedings of the 16th International Conference on Computational Linguistics* (COLING), I, Kopenhagen, 466–471.

KISS T., STRUNK J. (2006). Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics 32*: 485-525

LIN D., PANTEL P. (2001). DIRT discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

MA J., MOUNIER M., BLANCAFORT H., COUTO J., DE LOUPY C. (2011) LOL: Langage objet dédié à la programmation linguistique. *In Proceedings of TALN*.

MORIN E. (1999). Extraction de liens sémantiques entre termes à partir de corpus de textes techniques. *Thèse de doctorat*. Nantes.

SHANNON C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1), 3-55.

URIELI A. (2013). Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit. *PhD thesis*. Université de Toulouse II-Le Mirail.

Analyse des inférences pour la fouille d'opinion en chinois

LiYun Yan
ERTIM, Inalco, 75007 Paris, France
liyun.yan@inalco.fr

RÉSUMÉ

La fouille d'opinion est une activité essentielle pour la veille économique, facilitée par les réseaux sociaux et forums dédiés. L'analyse repose généralement sur des lexiques de sentiments. Pourtant, certaines opinions sont exprimées au moyen d'inférences. Dans cet article, nous proposons une classification des inférences utilisées en chinois dans des commentaires touristiques, à des fins de fouille d'opinion, selon trois niveaux d'analyse (réalisation sémantique, modalité de réalisation, et mode de production). Nous démontrons l'intérêt d'analyser les différents types d'inférence pour déterminer la polarité des opinions exprimées en corpus. Nous présentons également de premiers résultats fondés sur des plongements lexicaux.

ABSTRACT

Analysis of Inferences in Chinese for Opinion Mining

Opinion mining is an essential activity for economic watch, made easier by social networks and ad hoc forums. The analysis generally relies on lexicon of sentiments. Nevertheless, some opinions are expressed through inferences. In this paper, we propose a classification of inferences used in Chinese in tourist comments, for an opinion mining task, based on three levels of analysis (semantic realization, modality of realization and production mode). We proved the interest to analyze the distinct types of inferences to identify the polarity of opinions expressed in corpora. We also present some results based on word embeddings.

MOTS-CLÉS : Inférences, fouille d'opinion, polarité.

KEYWORDS: Inferences, Opinion Mining, Polarity.

1 Introduction

L'essor d'Internet permet aux utilisateurs d'échanger facilement leurs opinions et sentiments sur divers aspects de la vie quotidienne. Cette possibilité d'expression rapide constitue un enjeu de veille pour les entreprises (étude de la réputation, avis de satisfaction clientèle, etc.) et modifie également le mode de pensée des utilisateurs, soit par la possibilité de laisser un nombre élevé de commentaires de peu d'intérêt, soit par la possibilité de se retrancher derrière un commentaire anonyme pour exprimer un avis négatif (que l'absence d'anonymat n'aurait pas permis). Les messages laissés par les anciens clients témoignent de différences culturelles et sociales, dans le choix des critères d'évaluation (taille des chambres, présence d'équipements et services dans l'hôtel), dans l'utilisation du vocabulaire (terme générique *vs* terme spécifique au domaine), et dans la manière d'exprimer une information (en particulier les éléments jugés négatifs). En raison du nombre élevé de commentaires disponibles

sur Internet, il est nécessaire de disposer d'outils automatiques de fouille d'opinion pour analyser le contenu et dégager les tendances exprimées. En outre, les commentaires contiennent une grande quantité d'inférences qui demandent une analyse plus profonde pour la fouille d'opinion. Dans cet article, nous présentons les différents types d'inférence (section 2). Puis, à partir de notre corpus (section 3), nous proposons une classification des types d'inférences fondée sur trois niveaux d'analyse et mettons en évidence les variantes linguistiques en chinois. Enfin, nous présentons les résultats que nous obtenons en corpus (section 4) et présentons les résultats de plongements lexicaux appliqués en corpus pour mettre en évidence la relation des éléments d'une inférence dans un espace vectoriel.

2 État de l'art

2.1 Définition

Une inférence est une « opération par laquelle on passe d'une assertion considérée comme vraie à une autre assertion au moyen d'un système de règles qui rend cette deuxième assertion également vraie » (Larousse). Dans une approche déductive, ces règles permettent d'identifier la vérité d'une proposition à partir d'une ou plusieurs propositions prises en entrée. Dufaye (2001) considère que l'opération d'inférence consiste à poser un contenu non vérifié en prenant appui sur un contenu vérifié ou supposé vérifié.

Les inférences constituent également un processus d'interprétation, essentiel pour la compréhension du discours, dans la mesure où elles mettent en évidence des relations qui ne sont pas directement accessibles (Fayol, 2003). Gombert *et al.* (1992) considèrent que l'accès au sens ne provient pas directement du texte mais qu'il est construit par le lecteur, donc variable selon les individus en fonction de leurs connaissances. De même, Kispal (2008) indique que la compréhension des inférences est facilitée si le lecteur dispose de larges connaissances et qu'il partage le contexte culturel du texte. Dans les dialogues de la vie quotidienne, Beaupré (2009) estime qu'il n'existe aucune loi absolue, mais que les inférences reposent sur un processus de généralisation et de règles.

2.2 Types d'inférence

Bien qu'il existe un nombre important d'études sur les inférences, il n'existe pas de consensus sur une classification uniforme des différents types d'inférences (Lavigne, 2008), dans la mesure où tout travail de classification dépend à la fois du domaine scientifique et des objectifs visés.

Au niveau du document, Graesser *et al.* (1994) ont proposé trois types d'inférence pour expliquer le processus de compréhension : locales (au niveau de la phrase ou du paragraphe), globales (à l'échelle du document) et explicatives (proposant une reformulation).

Au niveau linguistique, Dufaye (2001) se fonde sur la théorie du sens élaborée par Peirce (1958) pour proposer une distinction des inférences fondée sur le mécanisme mis en œuvre mentalement : déduction, induction, et rétroduction. Ces trois types sont considérées par Peirce comme les trois figures du syllogisme (Deledalle, 1994). Les exemples de ces trois types sont présentés dans le tableau 4. Pour aboutir à une conclusion valide, la déduction suppose des prémisses valides alors que l'induction se fonde sur des probabilités. Selon le nombre de prémisses, la déduction peut être classifiée comme (i) l'inférence immédiate qui ne possède qu'une seule prémisse et que la

conclusion est aboutie via cette prémisse ou (ii) l'inférence médiale qui possède au moins 2 prémisses (Khemlani *et al.*, 2012). Les inférences rétroductives imposent de prendre en compte des connaissances antérieures (Deledalle, 1994).

Type d'inférence	Exemple	Traduction
Déduction	再来巴黎还会选择这里	On va encore choisir cet hôtel la prochaine fois à Paris.
Induction	唯一我不满的就是房间缺少一台咖啡机	La seule chose qui n'était pas satisfaite est qu'il manque une machine à café dans la chambre.
Rétroduction	洗澡地方对于小身材的亚洲人都有点拥挤，不知道歪果仁是怎么洗澡的在这么一个狭窄的地方。	La salle de bain est étroite, même pour des asiatiques de petite taille. Imaginez comment des étrangers peuvent prendre la douche dans un entroit aussi petit.

Tableau 1 – Exemples de déduction, induction et rétroduction

Duchêne (2008) distingue les inférences logiques des inférences pragmatiques. Les inférences logiques reposent sur un raisonnement formel et mettent en œuvre un processus logique alors que les inférences pragmatiques reposent sur un raisonnement inductif et s'appuient sur l'ensemble des connaissances acquises par un individu lors de ses expériences passées. Par exemple, il faut savoir que le Champ-de-Mars est à côté de la Tour Eiffel pour aboutir une conclusion positive pour le commentaire « *L'hôtel est à 5 min à pieds du Champ-de-Mars* ».

Doucy & Massous (2012) opèrent une distinction fondée sur le niveau d'analyse. Ils distinguent ainsi les inférences lexicales (la phrase en dehors de tout cadre énonciatif), les inférences énonciatives (un énoncé actualisé en contexte) et les inférences discursives (l'enchaînement cohérent de phrases). Les inférences lexicales et énonciatives s'inscrivent dans un continuum : les inférences lexicales construisent le sens à partir des structures prédicatives (prédicats et arguments) et les inférences énonciatives se fondent sur le sens ainsi construit pour l'inscrire dans une situation énonciative. Doucy & Massous (2012) soulignent également le fait que les connaissances extérieures au texte permettent de moduler le sens issu des inférences lexicales en apportant de nouvelles significations. Nous observons que cette opposition, fondée sur le niveau d'analyse, offre un cadre applicatif pertinent pour le traitement automatique des langues.

3 Matériel et méthodes

3.1 Description du corpus

Dans ce travail, nous nous intéressons aux inférences utilisées en chinois, dans un objectif de fouille d'opinion. A cet effet, nous avons rassemblé un corpus de commentaires postés sur trois sites par des touristes chinois en visite à Paris, sur la thématique de l'hébergement à Paris (en hôtel ou chez l'habitant) : Booking [1], Mafengwo [2] et TripAdvisor [3]. Les trois sites fournissent tous une plateforme

1. http://www.booking.com/
2. http://www.mafengwo.cn/
3. http://www.tripadvisor.fr/

pour que les utilisateurs puissent partager leurs propres expériences sur des séjours aux hôtels de Paris. Booking et TripAdvisor sont utilisés par des utilisateurs internationaux, en différentes langues, alors que Mafengwo est un site chinois, utilisé par des internautes sinophones (Chine continentale, Hong Kong, Taïwan). Bien que tous les utilisateurs de ces sites soient des voyageurs qui les utilisent pour planifier leurs voyages, nous observons une différence de classes d'âges sur le site Mafengwo, davantage fréquenté par de jeunes utilisateurs qui rédigent des blogs de voyage et postent des annonces pour des voyages collectifs.

Les commentaires sont rédigés en chinois simplifié et traditionnel. A chaque hôtel est associé des métadonnées comme son nom, son URL, une note globale et des notes pour chacune des propriétés (localisation, service, équipement, WIFI, propreté, confort, etc). Le corpus contient 1 776 hôtels sur Booking avec 27 043 commentaires dont la longueur moyenne est 36 caractères ; 2 017 hôtels sur Mafengwo avec 24 025 messages dont la longueur moyenne est 83 caractères par message. Les commentaires en chinois de Booking et Mafengwo sont tous écrits par des natifs, alors que les commentaires sur TripAdvisor sont mélangés car le site permet aux volontaires de traduire des messages en chinois. Il est donc possible d'observer sur le site TripAdvisor un phénomène culturel différent du public chinois dans les traductions d'expériences.

3.2 Analyse des inférences

A partir de notre corpus, nous avons manuellement étudié les différents types d'inférence disponibles. De cette analyse, nous avons établi une nouvelle classification des inférences que nous estimons pertinente pour notre tâche de fouille d'opinion en chinois. Nous avons abandonné la distinction ici non pertinente entre inférences médiates et immédiates car seules les inférences immédiates sont présentes dans notre corpus. Notre étude met en évidence trois niveaux d'analyse des inférences.

- réalisation sémantique : désigne comment se fait l'accès au sens exprimé dans l'inférence (inférence logique, pragmatique, ou lexicale)
- modalité de réalisation : désigne le processus mental que le locuteur met en œuvre pour accéder au sens (déduction, induction ou rétroduction)
- mode de production : renvoie à la manière dont l'émetteur du message a produit l'inférence (inférence énonciative ou discursive)

3.3 Variantes linguistiques en chinois

En analysant le corpus, nous avons observé que des variantes chinoises jouent un rôle important dans l'analyse des inférences. Nous observons deux types de variantes : (i) la polysémie, et (ii) la conversion du chinois simplifié vers le traditionnel.

Wu & Hsieh (2010) et Sheng (2011) considèrent les variantes en chinois comme des caractères ayant des formes visuelles différentes, mais qui ont la même prononciation ou la même signification. En traitement automatique des langues, ces cas ne sont pas évidents car ils partagent le même code Unicode. Comme les termes chinois utilisent plus d'un caractère, il est difficile de distinguer ou d'extraire les informations. Par exemple, 行 (U+884C) désigne le terme « marcher » 行走(U+884C, U+8D70) mais aussi « banque » 银行(U+94F, U+884C) (Lu *et al.*, 2016). De même, le caractère 黄 renvoie à quatre sens : 黄色 « jaune », 姓黄 renvoie au nom de famille Huang, 这事儿黄了 signifie que « la chose a échoué », et 扫黄 « anti-pornographie ». Selon le CDNC (Chinese Domain Name

Consortium), environ 40 % des caractères ont des formes variantes, ce qui souligne l'importance de prendre en compte cet aspect dans notre analyse.

Le chinois simplifié et le chinois traditionnel sont toujours utilisés de nos jours. Le chinois simplifié est utilisé en Chine (RPC) alors que le chinois traditionnel est utilisé à Taïwan, Hongkong et Singapour. Comme les commentaires des forums mélangent le chinois simplifié et le chinois traditionnel, nous considérons également ce type de variantes, d'autant plus que différentes expressions peuvent être influencées selon les régions. Par exemple, Halpern (2006) relève que « taxi » est noté 出租汽车 en chinois simplifié, 計程車 en chinois traditionnel de Taïwan, et 的士 en chinois simplifié de Hong-Kong.

En analysant le corpus, nous avons fréquemment observé des variantes linguistiques sur des thématiques, des opinions et des objets. Il s'agit parfois de synonymes, mais aussi de variantes traditionnelles. Les variantes sont complexes à traiter car les commentaires postés sur Internet ne respectent pas strictement les traductions normalisées.

4 Résultats et discussion

4.1 Classification et combinaison des inférences

Nous renseignons dans le tableau 4 le nombre d'inférences pour chaque catégorie et donnons dans le tableau 3 des exemples pour chacun des types d'inférence que nous avons identifiés en corpus.

Niveau d'analyse	Type	Nombre
Réalisation sémantique	logique	36 (19,9 %)
	pragmatique	91 (50,3 %)
	lexical	54 (29,8 %)
Modalité de réalisation	induction	17 (17,5 %)
	déduction	75 (77,3 %)
	rétroduction	5 (5,2 %)
Mode de production	discursif	58 (52,3 %)
	énonciatif	53 (47,7 %)

Tableau 2 – Nombre d'inférences pour chacun des types définis

Le traitement des inférences est à la fois indispensable et complexe, mais utile pour la fouille d'opinion pour trois raisons principales.

Premièrement, l'objet d'une opinion n'est pas toujours explicite dans le messages d'un utilisateur. Par exemple, « proche de la Tour Eiffel » implique de manière sous-entendue une localisation positive de l'hébergement, dans un contexte touristique. La proximité d'une station de métro est déjà plus complexe à interpréter. Cette localisation est-elle positive du point de vue de l'accès aux transports, ou négative en raison des nuisances engendrées ?

Deuxièmement, il n'est pas toujours possible de dégager des indices forts pour repérer facilement les phrases qui contiennent des inférences. Les mots porteurs de sentiments ou les formes morpho-syntaxiques ne permettent pas une identification de manière certaine. Le lecteur doit alors mobiliser

Type	Exemple	Traduction	Polarité
logique, déduction, discursif	我见过最小的卫生间，跟飞机上的差不多	La plus petite salle de bain que j'ai rencontrée, presque comme en avion	négatif
logique, déduction, énonciatif	酒店装修严重影响客户	Les travaux de l'hôtel gênent beaucoup les clients	négatif
logique, induction, discursif	退房时让酒店帮忙叫了车去机场，但我觉得价格贵了，可能被宰了	On a commandé un taxi à l'hôtel au moment du check-out pour aller à l'aéroport, mais le prix était cher, c'était peut-être une anarque	négatif
logique, induction, énonciatif	前台只有一个人，非常忙，每次都要排队等。	Il n'y a qu'une personne à l'accueil qui est très occupée, il faut faire la queue chaque fois	négatif
logique, rétroduction, discursif	巴黎人干什么都漫不经心，应该放在房间的手机到退房都没给，每天都说第二天。	Les parisiens font n'importe quoi. On n'a pas eu accès au téléphone dans la chambre jusqu'au moment du départ. Tous les jours on nous a dit qu'on nous le donnerait le lendemain	négatif
pragmatique, déduction, discursif	距离凯旋门约500米	L'Arc de Triomphe est à environ 500m	positif
pragmatique, déduction, énonciatif	离地铁站很近	proche du métro	positif
pragmatique, déduction, discursif	但铁塔景观并不理想，只能看到一些铁塔尖	Cependant, la vue de la Tour Eiffel n'est pas idéale, on voit seulement le bout de la pointe	négatif
pragmatique, induction, énonciatif	唯一我不满的就是房间缺少一台咖啡机。	La seule chose qui n'est pas satisfaisante est qu'il manque une machine à café dans la chambre.	négatif
pragmatique, induction, discursif	旁边有家乐福	Carrefour City à côté	positif
pragmatique, rétroduction, discursif	洗澡地方对于小身材的亚洲人都有点拥挤，不知道歪果仁是怎么洗澡的在这么一个狭窄的地方。	La salle de bain est étroite, même pour des asiatiques de petite taille. Imaginez comment des étrangers peuvent prendre la douche dans un entroit aussi petit	négatif
lexical	埃菲尔铁塔	Tour Eiffel	positif

Tableau 3 – Exemples des combinaisons d'inférences

des connaissances personnelles sur le monde et des compétences d'ordre linguistique pour décoder ces inférences. Ce travail se révèle encore plus complexe pour une machine, même en mobilisant des moyens du traitement automatique des langues.

Troisièmement, pour le domaine spécifique de l'hôtellerie, des inférences sont aussi représentées par le lexique du tourisme et des noms propres. Le traitement du lexique spécifique fait partie de l'analyse des inférences.

Dans la première série (logique-pragmatique-lexical), nous relevons que ces trois types peuvent

apparaître indépendamment de tout autre sous-type d'inférence (c'est-à-dire, sans combinaison avec le sous-type énonciatif ou discursif, ni avec un sous-type déduction, induction ou rétroduction). La forte proportion d'inférences de type « pragmatique » (50,3 % des inférences identifiées dans notre corpus) met en évidence l'intérêt de prendre en compte les informations culturelles pour effectuer une fouille d'opinion en chinois. Par ailleurs, 29,8 % des inférences sont lexicales. Il est ainsi nécessaire d'établir un lexique des termes utilisés dans le domaine touristique ou indicateur de sentiments. Du point de vue de la combinaison des inférences, la présence d'un seul type représente seulement 37,1 % des cas, alors que la combinaison de trois sous-types concerne jusqu'à 50 % des cas.

Dans notre corpus, toutes les inférences que nous avons identifiées expriment une opinion pour laquelle nous pouvons déterminer la polarité. Cela démontre que l'analyse des inférences est un enjeu important pour la fouille d'opinion en chinois.

4.2 Plongements lexicaux

Comme il n'existe pas d'indice fort pour identifier automatiquement les inférences, nous avons essayé de les traiter au moyen d'un apprentissage automatique. Dans cet article, nous avons utilisé un modèle Word2Vec qui permet de classer les similarités d'un mot candidat dans un espace vectoriel même sans des étiquettes grammaticales.

4.2.1 Protocole expérimental

A partir d'un corpus de 1 238 989 tokens, segmenté par l'outil jieba [4] (sans charger des dictionnaires extérieurs), nous avons extrait les 2000 premiers termes les plus fréquents. A l'aide du module gensim [5] de Python, nous avons ensuite entraîné un modèle Word2Vec. En ce qui concerne les paramètres, nous avons défini que le seuil de fréquence, la longueur de la fenêtre et la taille de dimension sont respectivement 5, 5 et 400. Enfin, nous avons retenu les 50 mots cibles les plus similaires à chaque candidat. Une partie des résultats est présentée dans le tableau 4.

4.2.2 Analyse des résultats

Nous effectuons les constatations suivantes :

— Les mots cibles qui constituent une inférence avec le mot candidat ne sont pas toujours les plus proches dans un espace vectoriel. Cette observation explique pour quelle raison il n'est pas évident d'identifier les inférences en corpus. Par exemple, les mots les plus proches de 地理位置 (localisation) sont 地段 (secteur, 0,933), 环境 (environnement, 0,736), 治安 (sécurité publique, 0,710), et 景色 (vue, 0,603). Ils correspondent aux éléments fondamentaux d'une localisation. Mais les mots qui permettent d'identifier une inférence apparaissent également dans cette liste : Montparnasse (0,574), 凯旋门 (Arc de Triomphe, 0,569), 地铁口 (entrée du métro, 0,563) et 卢浮宫 (Louvre, 0,553). Le lien entre localisation et ces mots de cible constitue une inférence pragmatique. Par exemple, la phrase 酒店地理位置在凯旋门旁边 (« *La localisation de l'hôtel est à côté de l'Arc de Triomphe* ») combine des inférences

4. https://github.com/fxsjy/jieba
5. https://pypi.python.org/pypi/gensim

N°	Token	Voisins distributionnels
1	前台 (accueil)	店员(personnel) 0,725 [...] 英语(personnel) 0,607 [...] 笑容 (sourire) 0,547) [...]
2	房间 (chambre)	屋子 (chambre) 0,741 [...] 卫生间 (toilette) 0,660 [...] 面积 (surface) 0,589 [...]
3	方便 (pratique)	便利 (pratique) 0,880 [...] 好找 (facile à trouver) 0,562 [...] 公交 (bus) 0,432 [...]
4	地铁站 (station du métro)	地铁口 (entrée du métro) 0,928 [...] 红磨坊 (moulin rouge) 0,724 [...] 巴黎圣母院 (Notre Dame de Paris)0,642 [...] 家乐福(Carrefour) 0,567 [...]
5	电梯 (ascenseur)	楼梯 (escalier) 0,721 [...] 行李箱 (valise) 0,575 [...] 缺点 (défaut) 0,451 [...]
6	免费 (gratuit)	下午茶 (goûter) 0,649 [...] 大厅 (hall) 0,621 [...] 打印机 (imprimante) 0,562 [...] 电热水壶 (bouilloire) 0,547 [...]
7	工作人员 (personnel)	员工 (employé) 0,928 服务员 (serveur) 0,910 店员 (vendeur) 0,861 服务生 (serveur) 0,801 人员 (personnel) 0,735 [...]

Tableau 4 – Exemples des voisins distributionnels pour quelques candidats

pragmatique (besoin de connaissances extérieures au texte), énonciative (dans une situation énonciative) et déductive (avec une prémisse solide afin de définir une polarité positive).
Ce genre de lien est fréquent entre le mot candidat et ses voisins distributionnels. Le mot candidat 前台 (« accueil », exemple 1) dans le tableau 4, constitue une inférence à la fois pragmatique et énonciative avec son voisin 英语 (« anglais »[6]). Il en est de même pour 免费 (« gratuit ») et son voisin 水壶 (« bouilloire », exemple 6) qui représente un stéréotype des Chinois qui ont l'habitude de boire de l'eau chaude.
Ces voisins distributionnels n'apparaissent cependant pas parmi les premiers (par score décroissant), mais à partir de la dixième place dans la liste de voisins. Nous considérons que cela fournit néanmoins un indice pour l'identification des inférences dans l'étape suivante de notre recherche.

— La méthode des plongements lexicaux donne un moyen d'établir une base de lexique spécifique et de regrouper des variantes dans l'hôtellerie en chinois. Par exemple, la liste des similarités du candidat 埃菲尔铁塔 (« Tour Eiffel ») contient quasiment tous les sites à Paris, alors que la liste du candidat 工作人员 (« personnel ») regroupe toutes les variantes des métiers de l'hôtellerie, ce qui est listé dans la 7e ligne du tableau 4. Cela correspond aussi aux besoins de traiter des variantes linguistiques pour l'analyse des inférences.

— La plupart des résultats montre une relation forte entre le mot candidat et les voisins associés, qui constituent des inférences pragmatiques ou lexicales. Ce phénomène explique également la grande proportion de l'inférence pragmatique et lexicale observée dans la partie précédente.

6. Le fait que l'anglais soit parlé à l'accueil de l'hôtel constitue un élément positif pour les touristes chinois en visite à Paris. Il est donc normal de trouver ce qualificatif comme voisin distributionnel du terme « accueil ».

5 Conclusion

En étudiant des recherches existantes concernant différents types d'inférence, nous avons constaté que l'inférence est une question peu traitée dans la fouille d'opinion.Dans cet article, nous avons réalisé une analyse des inférences dans un corpus de commentaires touristiques rédigés en chinois sur l'hébergement à Paris. De cette analyse, nous avons mis en évidence la complexité et la nécessité du traitement des inférences pour la fouille d'opinion en chinois. Ces tâches sont compliquées car il n'existe pas de méthode optimale pour identifier facilement les inférences pragmatiques. D'autre part, nous avons observé que la répartition est irrégulière ; une inférence relève de un à trois sous-types d'une part, et la répartition entre sous-types n'est pas homogène. Cependant, nous considérons que la prise en compte des inférences offre une piste pertinente pour réaliser la fouille d'opinion en chinois dans un domaine spécifique. Nos travaux futurs vont consister à intégrer l'analyse des inférences comme caractéristiques dans les algorithmes d'apprentissages statistiques tels que les plongements lexicaux (Word2Vec), de manière à réaliser une fouille d'opinion. Aussi, nous allons ajouter un prétraitement de normalisation pour éviter les erreurs de translittération.

Remerciements

J'adresse mes remerciements à M. Valette et M. Grouin, mes directeurs de thèse, pour leur précieuse aide à la relecture et à la correction de l'article.

Références

BEAUPRÉ S. (2009). L'approche dialectique pragmatique dans l'analyse des arguments. Master's thesis, UQAM, Montréal, Canada.

DELEDALLE G. (1994). Charles S. Peirce. les ruptures épistémiologiques et les nouveaux paradigmes. *Travaux du Centre de Recherches Sémiologiques*, **62**.

DOUCY G. & MASSOUS T. (2012). Sémantique inférentielle et compréhension des verbatim clients. In *Congrès Mondial de Linguistique Française*, volume 1.

DUCHÊNE A. (2008). Les inférences dans la communication : cadre théorique général. In *Actes de Rééducation orthophonique*, number 234. Fédération Nationale des Orthophonistes.

DUFAYE L. (2001). Les modaux et la négation en anglais contemporain. In *Cahiers de Recherche*. Ophrys.

FAYOL M. (2003). La compréhension : évaluation, difficultés et interventions. In *Actes de Conférence de Consensus*, Paris.

GOMBERT J.-E., FAYOL M., ZAGAR D., LECOCQ P. & SPRENGER-CHAROLLES L. (1992). *Psychologie cognitive de la lecture*.

GRAESSER A. C., SINGER M. & TRABASSO T. (1994). Constructing inferences during narrative text compréhension. *Psychological Review*, **101**(3).

HALPERN J. (2006). The role of lexical resources in CJK natural language processing. In *Proc of Multilingual Language Resources and Interoperability Work*, p. 9–16, Sydney, Australia.

KHEMLANI S., TRAFTON J. G., LOTSTEIN M. & JOHNSON-LAIRD P. N. (2012). A process model of immediate inferences. p. 151–156.

KISPAL A. (2008). *Effective Teaching of Inference Skills for Reading*. Rapport interne, Research Report DCSF-RR031.

LAVIGNE J. (2008). *Les mécanismes d'inférence en lecture chez les élèves de sixième année du primaire*. PhD thesis, Université Laval, Québec, Canada.

LU Y., ZHANG Y. & JI D. (2016). Multi-prototype chinese character embedding. In *Proc of LREC*, Portorož, Slovenia.

PEIRCE C. S. (1958). The collected papers of charles sanders peirce. In *Cambridge : Harvard University Press*, volume 1-6.

SHENG S. (2011). *Report on Chinese Variants in Internationalized Top-Level Domains*. Rapport interne, ICANN, Marina Del Rey, CA.

WU Y.-C. & HSIEH S.-K. (2010). PyCWN : a python module for Chinese Wordnet. In *Proc of COLING, Demo*, p. 5–8, Beijing, China.

Analyse des noms agentifs dans les espaces vectoriels distributionnels

Marine Wauquier[1]
(1) CLLE, CNRS & Université de Toulouse, 5 Allées Antonio Machado, 31058 Toulouse, France
marine.wauquier@univ-tlse2.fr

RÉSUMÉ

Notre étude s'inscrit dans le cadre d'une thèse ayant pour but d'exploiter les modèles distributionnels pour décrire sémantiquement des classes de mots définies selon des critères morphologiques. Nous utilisons des indices morphologiques et formels fournis par une base lexicale pour cibler les noms agentifs déverbaux construits par suffixation en *-eur*. Nous montrons qu'il est possible de constituer un représentant prototypique de la classe sémantique des noms agentifs en *-eur* dans les modèles distributionnels. L'étude de ce représentant met en évidence que l'information sémantique véhiculée par le suffixe varie en fonction du corpus d'étude et du degré de lexicalisation des dérivés.

ABSTRACT

Analysis of agent nouns in vector space models.

This experiment is part of a PhD thesis, the purpose of which is the semantic study through vector space models of word classes defined according to morphological criteria. We make use of morphological and formal clues given by a lexical database in order to target deverbal agent nouns derived with the french suffix *-eur*. We show that it is possible to build a prototypical representative of the agent noun semantic class in VSMs. The analysis of this representative highlights that the semantic instruction conveyed by the suffix varies depending on the corpus studied and the degree of lexicalization of the derivatives.

MOTS-CLÉS : Sémantique distributionnelle, sémantique lexicale, morphologie, linguistique de corpus.

KEYWORDS: Distributional semantics, lexical semantics, morphology, corpora linguistics.

1 Introduction

Dans le cadre de notre thèse, nous nous intéressons à la confrontation des modèles distributionnels à des catégories sémantiques établies selon des critères linguistiques, et en particulier morphologiques (noms d'agent, noms d'action, etc.) : notre objectif est d'utiliser les espaces vectoriels pour explorer la sémantique de ces catégories. Les outils distributionnels sont arrivés à maturité (Fabre & Lenci, 2015) et sont plébiscités pour leurs performances dans diverses tâches linguistiques (Kulkarni *et al.*, 2015; Verhoeven *et al.*, 2012). Mais les travaux se proposant de comparer sémantiquement des mots construits morphologiquement à l'aide d'outils d'analyse distributionnelle automatique sont à notre connaissance encore peu nombreux. Des travaux se sont par exemple penchsés sur l'étude des dérivés masculins et féminins en allemand, et ont montré que le genre des dérivés a un impact sur la distance distributionnelle (Zeller *et al.*, 2014). D'autres travaux ont quant à eux cherché à comparer

des procédés de nominalisation concurrents en allemand et ont montré qu'ils étaient distincts sur le plan distributionnel (Varvara *et al.*, 2016). Enfin, des travaux ont pour leur part exploité des indices distributionnels pour l'entraînement de classifieurs automatiques visant à distinguer les lectures événementielles des noms d'action polysémiques anglais en *-ment* (Lapesa *et al.*, a paraître).

Dans l'expérimentation que nous présentons plus spécifiquement dans cet article, nous nous intéressons aux noms agentifs déverbaux en *-eur* comme *chanteur* et *détecteur*, respectivement dérivés de *chanter* et *détecter*. Nous utilisons une ressource morphologique dérivationnelle, Lexeur, pour cibler ces déverbaux. Nous accédons à l'information sémantique véhiculée par ces déverbaux dans les corpus, et nous évaluons l'influence de deux paramètres, le corpus et le degré de lexicalisation (ou au contraire de nouveauté) du mot dérivé, sur la représentation distributionnelle de cette classe sémantique. Nous passons pour cela par une représentation prototypique que nous calculons à partir des membres de la classe sémantique. Nous montrons que le représentant prototypique de la classe des noms déverbaux en *-eur* que nous construisons véhicule un sens agentif, dans une acception relativement large, qui varie en fonction du corpus choisi et du niveau de lexicalisation de la classe considérée.

Nous détaillons dans un premier temps le dispositif expérimental que nous utilisons dans le cadre de cette étude (section 2). Nous présentons ensuite la démarche que nous avons suivie pour représenter sur le plan distributionnel la classe sémantique des agentifs déverbaux en *-eur* et les premières observations que l'on en tire, en nous penchant notamment sur la variation du premier paramètre que représentent les corpus (section 3). Nous évaluons ensuite l'influence de la lexicalisation en reprenant l'expérimentation avec de nouveaux paramètres (section 4). Enfin, nous discutons de ces résultats et nous évoquons les pistes de travail qu'ils ébauchent (section 5).

2 Dispositif expérimental

Nous faisons le choix d'utiliser conjointement un outil de sémantique distributionnelle, Word2Vec (Mikolov *et al.*, 2013), et une ressource dérivationnelle, Lexeur. L'utilisation de Lexeur permet de croiser des connaissances expertes validées par des linguistes avec les représentations sémantiques fournies par Word2Vec. Cette ressource nous permet en effet d'exploiter le critère morphologique comme indicateur de la classe sémantique, information absente de la représentation vectorielle, et de contrôler la nature des unités lexicales que nous considérons dans la première partie de notre expérimentation.

2.1 Ressource lexicale

Lexeur [1] regroupe 5 974 noms agentifs en *-eur* ainsi qu'une partie de leur famille dérivationnelle. Grâce à une procédure d'annotation manuelle, chaque nom agentif en *-eur* a été associé à son équivalent féminin en *-euse* et/ou en *-rice*, à sa base (*Vb* verbale ou *Nb* nominale) et à une liste de noms processifs identifiés. Ces noms sont issus du *Trésor de la Langue Française* (désormais *TLFi*), et complétés par des attestations issues du web. Chaque lexème est étiqueté morphosyntaxiquement. Nous utilisons le terme de noms agentifs dans une acception relativement large, puisque Lexeur regroupe indistinctement des noms d'agent (*abatteur*) et des noms d'instrument (*abatteuse*). De la

1. Cette ressource a été constituée par l'équipe CLLE-ERSS(Hathout & Fabre, 2002). En attendant sa mise en ligne sur le site REDAC, elle sera envoyée à toute personne sur simple demande à nabil.hathout@univ-tlse2.fr.

même façon, Lexeur regroupe sans distinction des noms d'action à l'interprétation événementielle (*abattage*), résultative (*abattis*), ou encore stative (*abattement*). Les lexèmes intégrés à la ressource présentent en outre divers degrés de polysémie (*sculpture* est à la fois une activité et un résultat). Toutes les familles ne sont pas homogènes. Le tableau 1 illustre la variété des entrées de Lexeur.

Nom d'agent masculin	**Nom d'agent féminin**	**Base**	**Cat.**	**Autres dérivés**
abatteur/Ncms	abatteuse/Ncfs	abattre/Vmn	Vb	abat/Ncms abattement/Ncms abatture/Ncfs abattage/Ncms abattis/Ncms
endoscopeur/Ncms	endoscopeuse/Ncfs	Ø	Ø	endoscopie/Ncfs
sculpteur/Ncms	sculpteuse/Ncfs sculptrice/Ncfs	sculpter/Vmn	Vb	sculpture/Ncfs ; sculptage/Ncms
wheeleur/Ncms	wheeleuse/Ncfs	wheel/Ncms	Nb	Ø

TABLE 1 – Extrait de Lexeur

L'ensemble des entrées comporte un nom agentif féminin en *-euse* et/ou en *-rice*, mais l'on peut noter que le suffixe *-euse* est surreprésenté (4 542 contre 1 514). 22% des familles sont dépourvues de noms processifs, à l'image de *wheeleur*. Pour les autres, on dénombre en moyenne 1,47 nom d'action par famille, le nombre de noms d'action par entrée variant entre 1 et 8. Nous nous concentrons dans cette étude sur les familles dont la base est verbale, ce qui représente 78% des entrées de Lexeur.

2.2 Corpus

Nos observations sont issues pour l'essentiel du corpus *Wikipédia*, constitué de la version française de 2013 de l'encyclopédie en ligne du même nom. Il compte environ 255 millions de mots. Du fait de son caractère encyclopédique, ce corpus couvre des domaines très divers, du plus général au plus technique, et présente donc un vocabulaire varié. Il est de fait particulièrement adapté à notre ressource qui regroupe des lexèmes à la technicité tout aussi variable (*chanteur*, *extérorécepteur*).

Ces observations sont confrontées à deux corpus aux genres textuels et à la taille variables, *LM10* et *frWaC*, pour tester la stabilité de nos observations. Le corpus *LM10* est constitué des articles du journal *Le Monde* publiés entre 1991 et 2000. Il contient environ 200 millions de mots. Le corpus *frWaC* est quant à lui composé de pages de sites internet appartenant au domaine .fr, pour un total de 1,3 milliard de mots. Nous avons choisi ce dernier corpus pour sa taille et pour les usages plus récents qu'il propose.

Les trois corpus ont été au préalable lemmatisés grâce à l'analyseur syntaxique Talismane (Urieli, 2013). Lorsque Talismane ne parvient pas à identifier le lemme d'une forme donnée, la forme est conservée. Cela explique la présence de certaines formes fléchies dans les résultats que nous présentons dans la suite de cette étude.

2.3 Espace vectoriel

Nous construisons les représentations distributionnelles des mots à l'aide de Word2Vec (Mikolov *et al.*, 2013). Nous nous basons dans cette étude sur l'examen de cooccurrences lexicales dans une fenêtre de 5 mots, sans prise en compte des relations syntaxiques. Word2Vec calcule le score de proximité de deux vecteurs, compris entre 0 (proximité nulle) et 1 (proximité maximale), sur la base du cosinus des vecteurs. Nous faisons le choix d'utiliser les paramètres par défaut [2] de Word2Vec, à savoir l'architecture CBOW, l'algorithme d'entraînement Negative Sampling, un seuil minimum de fréquence de 5, un seuil de sous-échantillonnage des mots fréquents de 10^{-3}, une taille de fenêtre maximale de 5, et comme nombre de dimensions des vecteurs 100. Nous construisons une matrice par corpus. Du fait des paramètres de l'outil, les mots issus de Lexeur et représentés dans les modèles vont donc varier en fonction du corpus. Le tableau 2 donne le nombre de noms agentifs présents dans les modèles.

	Wikipédia	***LM10***	***frWaC***
Masculin	1 334	1 147	1 444
Féminin	329	220	475

TABLE 2 – Nombre de noms agentifs issus de Lexeur et pris en compte en fonction du suffixe de constitution et du corpus

2.4 Construction d'un vecteur moyen

Notre objectif est de représenter de façon prototypique la classe des noms agentifs dans un corpus donné. Nous considérons ici la notion de prototype dans l'idée d'une catégorisation graduelle (Kleiber, 1990) : l'appartenance d'un élément à la catégorie est estimée selon son degré de ressemblance avec l'élément prototypique.

Nous abordons la notion de dérivé prototypique par le biais du dérivé moyen, dont nous définissons la représentation comme étant la moyenne des représentations des mots formés à partir de ce suffixe. Nous traduisons cela sur le plan vectoriel par la construction d'un barycentre, approche notamment utilisée dans des travaux sur la prédication verbale (Kintsch, 2001). Cela revient donc à calculer le vecteur $\overrightarrow{SUFF}$ du dérivé prototypique d'un suffixe *suff* comme la moyenne des vecteurs $\overrightarrow{Nsuff_i}$ des mots porteurs du suffixe tel qu'illustré en (1).

$$\overrightarrow{SUFF} = \frac{\sum_{i=1}^{n} \overrightarrow{Nsuff_i}}{n} \quad (1)$$

Nous ne prenons en compte que les vecteurs des mots présents dans Lexeur pour éviter de considérer des mots porteurs du suffixe correspondant mais n'appartenant pas à la catégorie sémantique visée (comme *fleur* pour *-eur*).

2. En réponse à la remarque d'un relecteur, nous discutons ce choix dans la section 5.

3 Représentation prototypique des noms agentifs en *-eur*

Le vecteur que nous construisons est abstrait car il agrège les informations des vecteurs qu'il moyenne. Il n'est pas la représentation d'un mot instancié dans le corpus. La question de son interprétation et de l'accès au sens qu'il véhicule se pose. À ce stade de la thèse, nous faisons le choix de passer par l'analyse des voisins distributionnels de ce vecteur théorique pour appréhender l'information sémantique qu'il véhicule. Nous favorisons une interprétation de nature qualitative, en nous limitant aux 50 premiers voisins. Pour le suffixe *-eur*, le 50 voisins les plus proches du dérivé prototypique calculé pour chacun de nos corpus sont présentés dans le tableau 3.

Wikipédia	réparateur - sèche-cheveux - soudeur - armurier - minuteur - wattman - conducteur - laborantin - machiniste - mécanicien - plombier - tournevis - stéthoscope - client - ventilateur - treuil - allumeur - mécano - coursier - déménageur - manomètre - aspirateur - soigneur - extincteur - vendeur - installateur - toiletteur - mélangeur - cric - ampèremètre - goniomètre - débogueur - technicien - ramasse-miettes - contacteur - descendeur - dépresseur - tune-o-matic - leurre - télérupteur - coupe-ongles - égoutier - microphone - juge-arbitre - opticien - nettoyeur - adaptateur - grappin - détecteur - ordinateur
LM10	ramoneur - bricoleur - toqué - alchimiste - chiot - nounours - moujik - magicien - ornithologue - matou - dragueur - bidouilleur - tâcheron - ludion - garagiste - fêlé - cinglé - comparse - imitateur - frelon - aventurier - coursier - barman - croque-mort - garnement - bouledogue - loubard - charretier - gandin - fripon - baroudeur - rouquin - coifeur - julot - boxeur - arnaqueur - malfrat - voyou - écuyer - prestidigitateur - moussaillon - cuisinier - sarret - puncheur - fêtard - camelot - afabulateur - braconnier - canari - garçon
frWaC	guindeau - perceur - surgrip - ferrailleur - rectifieur - multitours - coupe-papier - suiveur - basculeur - rilsan - servo-moteur - coupe-circuit - r-core - tromblon - palpeur - capsuleuses - accessoiriste - coupe-cigares - coutelas - marteau-pilon - talkie-walkie - rabot - cintreuses - pantographe - soudures - filin - emballeur - aspirateurs - petzl - montiss - cartillier - cintrier - batteur - mandrin - tuyautage - warbird - grignoteur - graisseur - perceuses - humbucker - elevateurs - gehennas - incorporateur - sebicape - dessouder - agitateur - encliquetables - avant-train - haute-pression - cymbales

TABLE 3 – 50 premiers voisins des dérivés prototypiques agentifs masculins en *-eur* dans les corpus *Wikipédia*, *LM10* et *frWaC*

La morphologie dérivationnelles se caractérise par une relation étroite entre la forme et le sens. Ces listes de voisins nous livrent des informations relatives à ces deux aspects.

Sur le plan sémantique, si l'on considère tout d'abord le corpus *Wikipédia*, on constate que les voisins du dérivé prototypique en *-eur* sont des noms de métier (*réparateur*, *armurier*), à raison de 44%, ou des noms d'instrument (*minuteur*, *coupe-ongles*), à raison de 46%. Dans le cas du corpus *LM10*, les 50 premiers voisins du dérivé prototypique en *-eur* sont majoritairement agentifs, à raison de 82% de noms de métiers (*ramoneur*, *charretier*) ou de noms d'humains (*garçon*, *comparse*). On retrouve par ailleurs 12% de noms d'animaux (*chiot*, *canari*). Enfin, parmi les 6% de voisins restants, on retrouve 2 noms propres, *sarret* et *camelot*, et un nom d'artefact ayant quasiment un statut d'instrument,

ludion. Pour ces 2 corpus, le dérivé prototypique semble donc bien capter le sens d'agentivité lié à la suffixation en *-eur*. On notera cependant que le passage d'un corpus encyclopédique à un corpus journalistique semble jouer sur la représentation de l'agentivité, perdant sa facette instrumentale.

Les résultats sont plus variés, et plus difficiles à interpréter, dans le cas du corpus *frWaC*. On est frappé par le caractère très spécifique du vocabulaire mis au jour : il s'agit de formes rares à la fréquence en moyenne plus faible (cf. tableau 5), pour certaines d'entre elles fléchies. On dénombre sur les 50 premiers voisins 52% d'instruments (*guindeau*, *coupe-cigares*), 16% d'agents (*ferrailleur*, *emballeur*) et 10% de noms à la double interprétation agentive et instrumentale (*batteur*). Parmi les 22% restants, on retrouve 4 entités nommées, dont 2 noms de marques d'outillage (*Petzl* et *Montiss*) et 2 noms propres (*Cartillier* et *Gehennas*). On constate par ailleurs la présence de 2 adjectifs (*encliquetables*, *haute-pression*), de 4 artefacts n'ayant pas clairement un statut d'instrument (*surgrip*, *soudures*) et d'un verbe (*dessouder*). On retrouve donc le sens d'agent et d'instrument que l'on avait déjà pour le corpus *Wikipédia*, mais de façon moins marquée.

Sur le plan formel, on constate que 56% des voisins du dérivé moyen ne sont pas suffixés en *-eur* dans le corpus *Wikipédia*. La tendance se confirme dans les corpus *LM10* et *frWaC* puisque respectivement 78% et 50% des voisins ne sont pas porteurs du suffixe *-eur*. On retrouve ainsi par exemple des dérivés suffixés en *-mètre*, *-ier*, ou encore *-ien*, trois suffixes qui forment néanmoins eux aussi des noms d'agent ou d'instrument.

Nous donnons à titre de comparaison dans le tableau 4 les 50 premiers voisins des vecteurs construits de la même façon mais à partir des déverbaux issus de la colonne « Nom d'agent féminin » de Lexeur (cf. tableau 1).

L'analyse des 50 premiers voisins du dérivé agentif féminin prototypique en fonction des corpus montre que les trois vecteurs captent une nouvelle fois une notion d'agentivité. En effet, les voisins obtenus pour les trois corpus sont des noms de métier (*coiffeuse*, *stripteaseuse*), des mots rattachés à des sujets culturellement associés à la femme (*stiletto*, *manucure*) ou des patronymes de femmes (*Herzigova*, *Sorokina*). Cette agentivité incorpore ici des informations relatives au genre féminin, notamment une connotation négative (Wauquier *et al.*, 2018), et est bien distincte de l'agentivité observée pour les agentifs en *-eur*. On observe par ailleurs de nouveau une différenciation entre le corpus journalistique et les deux autres corpus. Les voisins obtenus pour les corpus *Wikipédia* et *frWaC* sont pour beaucoup des entités nommées, quand le corpus *LM10* en est exempt. L'analyse des voisins obtenus pour *frWaC* est une nouvelle fois rendue plus difficile du fait de la faible fréquence des formes liées au féminin, plus nette encore dans ce corpus (cf. tableau 5).

Ces résultats montrent une certaine hétérogénéité de la classe sémantique des noms agentifs en *-eur*. On y distingue ainsi plusieurs sous-classes déjà connues (Huyghe & Tribout, 2015), comme les noms de métiers, les noms d'instruments, ou encore des noms référentiels, mais aux comportements bien distincts sur le plan distributionnel.

4 Impact de la lexicalisation

La ressource Lexeur nous a été utile dans cette première étape pour garantir la validité des unités lexicales observées. À l'aune de sa couverture assez large, nous considérons dans la suite de cette étude que l'absence d'un nom agentif en *-eur* dans la ressource Lexeur est un indice du caractère néologique de cet agentif. Mais cette ressource a de fait l'inconvénient de limiter le vocabulaire

Wikipédia	herzigova - coiffeuse - venhard - naymark - manucure - vericel - sorokina - trulle - cover-girl - gitane - séménoff - chammah - comédienne - estragnat - yma - stroyberg - réju - tallier - soubrette - alycia - montalant - minouche - dartonne - ménine - metmer - rembauville - jitka - catzéflis - prepon - denarnaud - marie-olivier - tainsy - cuisinière - chauffeuse - anicée - serveuse - stripteaseuse - kajmak - laury - ballerine - barmaid - lunchlady - pierens -laparé - servantie - mammamia - stresi - irma - elfride - vendell
LM10	duègne - rousse - jolie - gitane - pulpeux - vamp - ravissant - bacchante - chatte - diablesse - boulotte - mignonne - allumeuse - madone - rockeuse - danseuse - parisienne - nymphomane - débutante - brune - mégère - lhamo - almée - ingénue - soubrette - véro - blonde - mamelue - pimbêche - adorable - femme-objet - femme-oiseau - garce - pétulant - servante-maîtresse - servante - dévergond - antillaise - trémière - courtisane - arnaqueuse - donzelle - nastassia - diva - guenon - chasseresse - junon - demi-mondaine - rieuse - belle-de-nuit
frWaC	kiraly-picot ouria - roitfeld - joustra - pezeril - extrado - montiss - punkette - gommettes - poupee - lainer - capsuleuses - spigarelli - pomagalski - bouvrain - prucnal - diffused - klinge - biboud - naomie - vitteaut - existais - corbery - raszewski - perleuse - taraud - baetz - planard - trouvain - wathier - gouttiere - loutch - robart - yomoshi - danner - cheriie - melodick - devraigne - plaignaud - stiletto - dheran - nallamoutou - poupée - morganne - turbulette - impertinante - marieras - chipette - carpenito - burada

TABLE 4 – 50 premiers voisins des dérivés prototypiques agentifs féminins en *-euse* et *-rice* dans les corpus *Wikipédia*, *LM10* et *frWaC*

	Wikipédia	***LM10***	***frWaC***
Masculin	814	451	163
Féminin	72	142	18

TABLE 5 – Fréquence moyenne des voisins des dérivés prototypiques en fonction du suffixe de constitution et du corpus

observable, en focalisant l'analyse sur les formes installées dans le lexique de la langue, et dites lexicalisées, puisque issues pour la majeure partie du *TLFi*. Leur sens a donc potentiellement évolué depuis leur formation, du fait de l'histoire de chaque mot. Nous pouvons notamment citer comme exemple le nom *échangeur*, dérivé du verbe *échanger*, qui a maintenant davantage le sens de système autoroutier que de personne ou d'instrument permettant d'échanger quelque chose. Or, le suffixe *-eur* est productif, et il produit encore des noms d'agents et d'instruments (Dubois, 1962; Dubois & Dubois-Charlier, 1999; Aronoff & Lindsay, 2014).

Nous faisons l'hypothèse que les formes les plus récentes obtenues par la suffixation en *-eur* sont sémantiquement plus transparentes vis-à-vis de l'instruction sémantique de leur verbe de base, à l'image de *bloggeur* et *blogger*. Nous souhaitons donc créer une représentation de ces agentifs néologiques, afin d'évaluer l'impact de la lexicalisation sur la classe sémantique des noms agentifs. Nous reproduisons pour cela la même expérience mais à partir de noms agentifs déverbaux suffixés en *-eur* qui ne sont pas présents dans Lexeur.

4.1 Repérage des noms agentifs en *-eur* néologiques

4.1.1 Extraction automatique de paires potentielles

Nous extrayons dans un premier temps des paires (N*eur*, V). Nous cherchons des paires, et pas simplement des noms en *-eur*, pour garantir que le nom en *-eur* est bien un nom déverbal. La procédure est la suivante : dans un premier temps, nous récupérons l'ensemble des 4 675 paires (N*eur*, V) présentes dans Lexeur. Pour chaque paire, un programme apprend la règle de transformation formelle liant le dérivé à sa base (Tanguy & Hathout, 2007). Dans un second temps, le programme parcourt l'ensemble des mots du vocabulaire du modèle distributionnel considéré. Le programme traite chaque mot finissant par la chaîne *-eur* comme un dérivé potentiel. Il lui attribue alors une base potentielle, à partir des règles qu'il a apprises précédemment. Un même dérivé peut éventuellement se voir attribuer plusieurs bases potentielles, à l'image de *superordinateur* qui est associé à *superordonner*, *superordiner* et *superordinater*. Nous nous retrouvons à ce stade avec 6 152 paires potentielles pour le corpus *Wikipédia*, contre 3 677 pour le corpus *LM10* et 10 665 pour le corpus *frWaC*. Dans un dernier temps, le programme élimine les paires dont la base potentielle est absente du modèle, soit parce qu'elle n'est pas assez fréquente, soit parce qu'elle n'existe pas. Nous obtenons 218 paires pour le corpus *Wikipédia*, contre 87 pour le corpus *LM10* et 726 pour le corpus *frWaC*.

4.1.2 Vérification manuelle des paires

Une étape de vérification manuelle s'ensuit pour ne conserver que les paires sémantiquement et formellement valides. Nous considérons comme valide une paire dont le premier élément est bien un nom agentif (agent ou instrument) en *-eur* dérivé d'un verbe, et dont le second élément est bien le verbe duquel est dérivé le nom agentif considéré, à l'image de *bloggeur* et *blogger*. Nous faisons donc le choix d'exclure les paires erronées selon les critères suivants :

— La base ne correspond pas à une forme verbale dans le corpus. Cela exclut des paires comme (*seigneur*, *seigner*), où *Seigner* est un nom propre.
— Le dérivé n'a pas d'emploi agentif dans le corpus. Cela exclut des paires comme (*sueur*, *suer*).
— Il s'agit d'une variante orthographique d'une paire présente dans Lexeur. Cela exclut les paires (*co-producteur*, *co-produire*) ou (*realisateur*, *realiser*) puisque les paires (*coproducteur*, *coproduire*) et (*réalisateur*, *réaliser*) sont déjà présentes dans Lexeur. Nous n'excluons cependant pas les paires comme (*coanimateur*, *coanimer*), malgré la présence dans Lexeur de la paire (*animateur*, *animer*), puisque la préfixation pourrait ici se traduire par une variation sémantique en corpus.
— Le verbe et son dérivé ne sont pas liés sémantiquement. Cela exclut des paires comme (*primeur*, *primer*).

À l'issue de cette vérification manuelle, nous obtenons 81 paires pour le corpus *Wikipédia*, 27 pour le corpus *LM10* et 169 pour le corpus *frWaC*.

4.2 Comparaison du comportement distributionnel

Nous avons d'abord comparé sur le plan distributionnel le comportement des paires (N*eur*, V) issues de Lexeur à celles issues du corpus. Puisque le nom agentif pour les paires néologiques est considéré

comme régulier, nous faisons l'hypothèse que le verbe et son déverbal en *-eur* seront sémantiquement plus proches que ne le seraient un verbe et son déverbal lexicalisé. Cela devrait donc se traduire, sur le plan distributionnel, par un score de proximité plus important pour les paires issues du corpus. Nous faisons par ailleurs l'hypothèse que cela s'appliquera à une majorité des paires considérées. Cela se traduirait par une dispersion moindre des paires et donc un écart type plus faible.

Nous faisons la moyenne des scores de proximité fournis par Word2Vec pour les paires issues de Lexeur et pour celles issues du corpus et nous comparons ces scores de proximité moyens. Nous calculons aussi l'écart type pour évaluer la dispersion de nos paires.

	frWaC		***Wikipédia***		***LM10***	
	Lexicalisation	Néologie	Lexicalisation	Néologie	Lexicalisation	Néologie
Proximité	0.293	**0.307**	0.271	**0.324**	0.262	**0.346**
Écart type	**0.171**	0.185	**0.165**	0.197	**0.163**	0.181

TABLE 6 – Score de proximité moyen et écart type entre le verbe et son nom agentif en *-eur* en fonction du corpus et du type de noms agentifs

Les résultats regroupés dans le tableau 6 montrent que le score de proximité entre le verbe et son déverbal en *-eur* est en moyenne plus élevé pour les paires à caractère néologique que pour les paires lexicalisées. Cela va donc dans le sens de notre hypothèse initiale d'une plus grande proximité liée à une plus grande transparence du dérivé. Le calcul du t-test montre que la différence est significative pour les corpus *Wikipédia* et *LM10* (p-value de 0.01 et 0.02), mais qu'elle ne l'est pas pour le corpus *frWaC* (p-value = 0.3). Par ailleurs, nous constatons que l'écart type est plus élevé pour les paires issues du corpus que pour les paires issues de Lexeur. Cela signifie que les verbes sont en moyenne plus proches de leur dérivé néologique, mais avec une dispersion plus importante.

Nous complétons notre observation par l'analyse de la distribution des paires (N*eur*, V) en fonction de leur score de proximité, selon leur degré de lexicalisation dans les trois corpus considérés (figure 1).

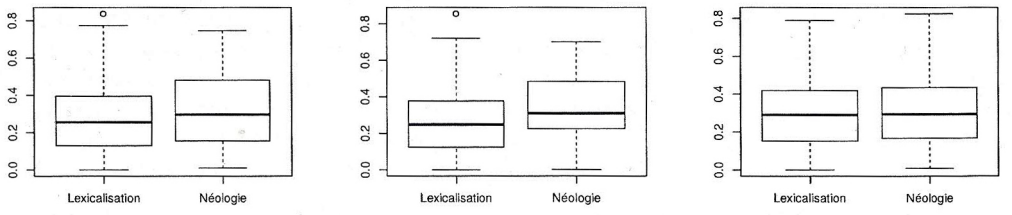


FIGURE 1 – Distribution des paires en fonction de leur score de proximité et du degré de lexicalisation du nom agentif. À gauche : corpus *Wikipédia*. Au centre : corpus *LM10*. À droite : corpus *frWaC*

On constate que la distribution des paires est similaire dans les corpus *Wikipédia* et *LM10*. On observe en effet un score de proximité médian et un score de proximité pour les interquartiles plus élevés pour les paires néologiques que pour les paires lexicalisées, et des extrêmes légèrement moins marqués. Ces observations vont une nouvelle fois dans le sens de notre hypothèse. Le corpus *frWaC* se caractérise quant à lui par une distribution quasi identique des paires quel que soit leur degré de lexicalisation. La seule variation observée concerne les valeurs extrêmes supérieures, qui semblent légèrement plus

élevées pour les paires néologiques. Ces résultats confirment le statut un peu à part du corpus *frWaC* dont nous analysons plus précisément les résultats en 4.3.

Une fois établies ces tendances générales, nous relevons quelques comportements particuliers dans l'ensemble des paires néologiques. Certaines présentent des scores de proximité très élevés (>0.7). C'est le cas de paires comme (*parseur*, *parser*) ou (*sampleur*, *sampler*) pour le corpus *frWaC*, (*parseur*, *parser*) et (*vocodeur*, *vocoder*) pour le corpus *Wikipédia*, ou (*rappeur*, *rapper*) pour le corpus *LM10*. Une analyse en corpus de ces différents lexèmes montre que les termes *parser*, *sampler*, *vocoder* ou *rapper* sont très majoritairement utilisés en tant que nom commun et non en tant que verbe, en lieu et place de leur équivalent en *-eur*. L'utilisation de ces anglicismes expliquent donc la forte proximité des formes de chaque paire.

On constate par ailleurs la présence de paires néologiques au score de proximité très faible (<0.05), là encore pour les trois corpus. On retrouve parmi elles des paires comme (*raveur*, *raver*) ou (*raideur*, *raider*) pour le corpus *frWaC*, (*desserviteur*, *desservir*) pour le corpus *Wikipédia* ou (*auditeur*, *auditer*) pour le corpus *LM10*. Une analyse des contextes d'apparition montre ainsi que *Raver* est principalement utilisé comme un nom propre. Enfin, les lexèmes les plus instanciés en corpus par les formes *raideur*, *desservir* ou *auditeur* ne correspondent pas aux familles sémantiques visées par les paires (N*eur*, V) auxquelles les formes appartiennent.

Dans les deux cas, que la proximité soit faible ou forte, cela met en exergue des paires problématiques. Un score de proximité en décalage par rapport à la tendance globale semble donc être un bon indice pour détecter les paires atypiques. On peut s'interroger sur la pertinence d'avoir conservé les paires comme (*parseur*, *parser*) évoquées précédemment. Ce choix est dû aux critères explicités dans la section 4.1.2 que nous avons appliqués de façon stricte, mais dont on voit ici les limites. Le fait d'avoir au moins une occurrence agentive ou verbale en corpus ne semble ainsi pas assez restrictif. De la même façon, le caractère néologique de certaines paires comme (*auditeur*, *auditer*) est à remettre en question. Ainsi, *auditeur* n'est pas un néologisme, et on le retrouve dans Lexeur dans deux familles, une liée au nom *audition*, et l'autre au nom *audit*. Si la paire (*auditeur*, *auditer*) n'est effectivement pas présente dans Lexeur, ce n'est pas dû à son caractère néologique, mais à un problème d'exhaustivité de la ressource.

En revanche, parmi les paires au comportement typique, on observe de nombreux cas de figure intéressants. On retrouve ainsi des paires comme (*uploadeur*, *uploader*), (*tchatteur*, *tchatter*), (*débogueur*, *déboguer*), (*bloggeur*, *blogger*) et (*multiplexeur*, *multiplexer*). Ces paires sont précisément celles qui nous intéressent puisqu'il s'agit bien ici de cas de dérivation récente d'agentifs à partir de verbe, où le sens du dérivé n'a pas encore évolué.

4.3 Observation des voisins des noms agentifs néologiques

Nous faisons le choix de nous concentrer plus précisément sur le corpus *frWaC* car c'est celui pour lequel nous avons le plus grand nombre de paires et donc de noms agentifs néologiques. Nous souhaitons par ailleurs nous pencher davantage sur les résultats *a priori* contre-intuitifs présentés en 4.2.

De la même façon que dans la section 3, nous créons le vecteur représentant la moyenne des vecteurs des noms agentifs en *-eur* récupérés, et nous en analysons les 50 premiers voisins (tableau 7). Nous les comparons avec les 50 premiers voisins du dérivé prototypique lexicalisé dans le corpus *frWaC* reportés dans la dernière partie du tableau 3.

quenc - mytheather - toneport - expandeur - electret - comptact - microcontroleur - le-gnou - handsonic' - webeditor - genlock - go-to - attiny - oehlbach - o-system - shamallows - easybox - mickeyfreestyler - aldila - atmega - whirlwind - r-core - audiounit - coprocesseur - mini-navigateur - frw - enhancer - selector - seeprog - serato - realtek - twido - wago-i - testeur - hwmonitor - multiprogrammateur - speedo - winup - pod - crystalin - textorm - modul - beeprog - yokogawa - modutils - dragonfly - sniffeur - electromatic - hammerhead - encodeur

TABLE 7 – 50 premiers voisins du dérivé néologique prototypique en *-eur* dans le corpus *frWaC*

Rappelons que dans le cas des 50 premiers voisins obtenus à partir des noms agentifs présents dans Lexeur, on dénombrait seulement 10% d'entités nommées (*Petzl*, *Cartillier*) contre 88% de lexèmes qui n'étaient pas des entités nommées (*surgrip*, *ferrailleur*, *encliquetables*). Les 2% restants correspondaient à la forme *Rilsan*, qui peut être utilisé comme une entité nommée ou un nom commun. Ces voisins ne relèvaient pas du champ sémantique des nouvelles technologies, à quelques exceptions près (*Gehennas*), mais davantage du monde industriel ou technique (*servo-moteur*, *tuyautage*).

Le comportement diffère grandement pour les voisins du dérivé néologique prototypique dans *frWaC*. La première chose que l'on constate est que les 50 premiers voisins du dérivé prototypique en *-eur* formés à partir des noms agentifs absents de Lexeur sont peu fréquents (*oehlbach*, *attiny*), avec une fréquence moyenne de 54 contre 163 pour les voisins obtenus à l'aide des déverbaux issus de Lexeur (tableau 5). L'interprétation de ces voisins est d'autant plus compliquée qu'il s'agit majoritairement d'entités nommées (à raison de 68%), pour lesquelles il est difficile de capter une homogénéité sémantique. On retrouve notamment des marques (*Yokogawa*), des logiciels (*quEnc*) ou encore des pseudonymes (*MickeyFreeStyler*). On ne compte *a contrario* que 26% de lexèmes qui ne sont pas des entités nommées (*webeditor*, *crystalin*). Enfin, les 6% restants sont des formes qui sont utilisées en corpus tantôt en tant qu'entité nommée, tantôt en tant que nom commun. Notons qu'une grande partie de ces noms dénotent des objets, informationnels ou matériels, ou des instruments. Tous les voisins sont néanmoins liés par le champ sémantique auquel ils se rapportent, celui de l'informatique, de l'électronique et des nouvelles technologies. Ces résultats soulignent l'hétérogénéité du corpus *frWaC* mis en lumière par la figure 1. Il est en effet quantitativement plus important, mais son contenu, plus bruité, est plus difficile à exploiter.

5 Discussion

Lors de cette expérimentation, nous avons pu comparer sur le plan distributionnel des représentants de la classe sémantique des noms agentifs déverbaux en *-eur*, en évaluant l'impact du corpus et de leur niveau de lexicalisation. Les résultats préliminaires sont encourageants puisque nous parvenons à capter la notion d'agentivité au travers d'un représentant prototypique. Nous avons vu que cette représentation permettait d'accéder à certaines composantes du sens de la classe de mots observée, puisqu'elle intégrait l'information liée au genre dans le cas de l'analyse des noms agentifs féminins en *-euse* et *-rice*, et qu'elle permettait même d'accéder à des connaissances du monde liées à la connotation de ces noms agentifs féminins (Dawes, 2003; Schafroth, 2001). Nous avons par ailleurs constaté que les déverbaux suffixés en *-eur* ne constituent pas une classe sémantique homogène dans les modèles Word2Vec utilisés dans cette étude. On observe ainsi la présence de plusieurs catégories au sein de la classe des noms agentifs en *-eur*, à savoir les noms d'instruments et les noms d'agents, se

divisant elles-mêmes en sous-classes ayant des comportements distincts. Enfin, nous avons travaillé sur des noms agentifs néologiques, et nous avons montré qu'ils étaient distributionnellement plus proches de leur base que ne le sont les noms agentifs lexicalisés dans les corpus *Wikipédia* et *LM10*. Nous avons cependant constaté que ce n'était pas le cas dans le corpus *frWaC*, soulignant le caractère hétérogène de ce corpus mis en avant dans le reste de cette étude. Cela nous invite à nous interroger sur l'utilisation de corpus de taille importante, mais peu contrôlés et donc potentiellement très bruités, pour faire de l'analyse linguistique fine.

Nous envisageons à ce titre de poursuivre notre travail sur l'anglais. Les corpus y sont plus conséquents, et nous pourrions exploiter les informations sémantiques, morphologiques et formelles de certaines ressources à la couverture importante (Fellbaum & Miller, 2003). Outre la question du contrôle du corpus, une piste méthodologique ébauchée par cette étude concerne la démarche de sélection de noms agentifs en *-eur* néologiques. Nous avons vu les limites de notre définition de la néologie, mais nous avons fait le choix de conserver toutes les paires que nous avions constituées du fait de leur nombre déjà relativement limité. Cela nous invite néanmoins à reconsidérer nos critères de filtrage. Un premier moyen serait de nous servir de la distribution pour évincer les paires atypiques, ou du moins appeler à leur vérification manuelle. Un second moyen, complémentaire du premier, consisterait en une annotation morphosyntaxique des formes en corpus. Cela permettrait de ne conserver que les paires dont les deux membres ont pour acception principale le lexème voulu (verbe ou nom commun), et d'évincer de potentiels anglicismes. La projection d'un lexique anglais permettrait également de filtrer les formes erronées.

Si nous avons pu vérifier la stabilité des observations à travers différents corpus, nous n'avons pas pris en compte d'autres paramètres liés au modèle. Nous avons fait le choix d'utiliser dans cette étude le modèle par défaut fourni par Word2Vec, en l'état, sans chercher dans un premier temps à intervenir sur ses hyperparamètres, comme le nombre de dimensions ou l'architecture. Nous souhaitons à ce stade de la thèse exploiter l'outil tel qu'il est dans le but d'orienter notre analyse linguistique. Malgré une analyse corpus par corpus et non globalisante des représentations prototypiques, nous devrions dans l'idéal moyenner les distances de plusieurs représentations distributionnelles (Antoniak & Mimno, 2018), mais nous n'avons pas pu réaliser ces expérimentations par manque de temps. Nous avons cependant reproduit la manipulation en fixant le nombre de dimensions des vecteurs à 300 sans que les résultats varient significativement. Dans cette optique, nous envisageons de comparer les différentes techniques distributionelles, et avons entamé la reprise de l'expérimentation avec l'outil fastText (Bojanowski *et al.*, 2016). Nous avons par ailleurs fait le choix dans cette étude de baser nos observations sur les 50 premiers voisins des vecteurs construits, mais nous envisageons à terme d'analyser de façon plus systématique les voisins des vecteurs construits en observant la répartition et la densité des noms d'agent dans ce voisinage.

Références

ANTONIAK M. & MIMNO D. (2018). Evaluating the Stability of Embedding-based Word Similarities. *Transactions of the Association for Computational Linguistics*, **6**, 107–119.

ARONOFF M. & LINDSAY M. (2014). Productivity, Blocking and Lexicalization. *The Oxford handbook of derivational morphology*, p. 67–83.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2016). Enriching Word Vectors with Subword Information. *arXiv preprint arXiv :1607.04606*.

DAWES E. (2003). La féminisation des titres et fonctions dans la francophonie : de la morphologie à l'idéologie. *Ethnologies*, **25**(2), 195–213.

DUBOIS J. (1962). *Étude sur la dérivation suffixale en français moderne et contemporain : essais d'interprétation des mouvements observés dans le domaine de la morphologie des mots construits*. Paris : Larousse.

DUBOIS J. & DUBOIS-CHARLIER F. (1999). *La dérivation suffixale en français*. Paris : Nathan.

FABRE C. & LENCI A. (2015). Distributional Semantics Today Introduction to the Special Issue. *Traitement Automatique des Langues*, **56**(2), 7–20.

FELLBAUM C. & MILLER G. A. (2003). Morphosemantic Links in Wordnet. *Traitement Automatique des Langues*, **44**(2), 69–80.

HATHOUT N. & FABRE C. (2002). Constitution et exploitation de lexiques de formes déverbales. *Communication aux Journées d'études sur les noms déverbaux. Silex, Université Lille*, **3**.

HUYGHE R. & TRIBOUT D. (2015). Noms d'agents et noms d'instruments : le cas des déverbaux en-eur. *Langue française*, (1), 99–112.

KINTSCH W. (2001). Predication. *Cognitive science*, **25**(2), 173–202.

KLEIBER G. (1990). *La sémantique du prototype : catégories et sens lexical*. PUF.

KULKARNI V., AL-RFOU R., PEROZZI B. & SKIENA S. (2015). Statistically Significant Detection of Linguistic Change. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, p. 625–635, Florence, Italy.

LAPESA G., KAWALETZ L., PLAG I., ANDREOU M., KISSELEW M. & PADO S. (à paraître). Disambiguation of Newly Derived Nominalizations in Context : A Distributional Semantics approach. (draft).

MIKOLOV T., CHAN K., CORRADO G. & DEAN J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of International Conference on Learning Representations (ICLR)*, Scottsdale, United States of America.

SCHAFROTH E. (2001). *Gender in French Structural Properties, Incongruences*, In *Gender Across Languages : The Linguistic Representation of Women and Men*, volume 3, p. 87–117. John Benjamins Publishing.

TANGUY L. & HATHOUT N. (2007). *Perl pour les linguistes*. TIC et Sciences Cognitives. Hermès sciences publications. Site d'accompagnement : http ://perl.linguistes.free.fr.

URIELI A. (2013). *Robust French Syntax Analysis : Reconciling Statistical Methods and Linguistic Knowledge in the Talismane Toolkit*. PhD thesis, Université de Toulouse II le Mirail.

VARVARA R., LAPESA G. & PADÓ S. (2016). Quantifying Regularity in Morphological Processes : An Ongoing Study on Nominalization in German. In *ESSLLI DSALT Workshop : Distributional Semantics and Semantic Theory*, Bolzano, Italy.

VERHOEVEN B., DAELEMANS W. & VAN HUYSSTEEN G. (2012). Classification of Noun-noun Compound Semantics in Dutch and Afrikaans. In *Proceedings of the 23rd Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, p. 121–125, Pretoria, South Africa.

WAUQUIER M., FABRE C. & HATHOUT N. (2018). Différenciation sémantique de dérivés morphologiques à l'aide de critères distributionnels. In *Congrès Mondial de Linguistique Française (CMLF)*, Mons, Belgique.

ZELLER B. D., PADÓ S. & SNAJDER J. (2014). Towards Semantic Validation of a Derivational Lexicon. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, p. 1728–1739, Dublin, Ireland.

Analyse formelle d'exigences en langue naturelle pour la conception de systèmes cyber-physiques

Aurélien Lamercerie[1]
(1) Univ Rennes, Inria, IRISA - UMR 6074, F-35000 Rennes, France
aurelien.lamercerie@inria.fr

RÉSUMÉ
Cet article explore la construction de représentations formelles d'énoncés en langue naturelle. Le passage d'un langage naturel à une représentation logique est réalisé avec un formalisme grammatical, reliant l'analyse syntaxique de l'énoncé à une représentation sémantique. Nous ciblons l'aspect comportemental des cahiers des charges pour les systèmes cyber-physiques, c'est-à-dire tout type de systèmes dans lesquels des composants logiciels interagissent étroitement avec un environnement physique. Dans ce cadre, l'enjeu serait d'apporter une aide au concepteur. Il s'agit de permettre de simuler et vérifier, par des méthodes automatiques ou assistées, des cahiers des charges "systèmes" exprimés en langue naturelle. Cet article présente des solutions existantes qui pourraient être combinées en vue de la résolution de la problématique exposée.

ABSTRACT
Formal analysis of natural language requirements for the design of cyber-physical systems

This paper focuses on the construction of formal representations of natural language texts. The mapping from a natural language to a logical representation is realized with a grammatical formalism, linking the syntactic analysis of the text to a semantic representation. We target the behavioral aspect of the specifications for cyber-physical systems, *ie* any type of system in which software components interact closely with a physical environment. In this way, the challenge would be to provide assistance to the designer. So, we could simulate and verify, by automatic or assisted methods, "systems" specifications expressed in natural language. This paper presents some existing contributions that could enable progress on this issue.

MOTS-CLÉS : Formalisme grammatical, Représentation sémantique, Grammaire catégorielle, Ingénierie des cahiers des charges, Système cyber-physique, Spécification modale.

KEYWORDS: Grammatical Formalism, Semantic Representation, Categorial Grammar, Requirements Engineering, Cyber-physical System, Modal Specification.

1 Introduction

L'objectif de cet article est de proposer les éléments d'une méthodologie permettant de transformer certains fragments de cahiers des charges, exprimés en langue naturelle, en spécifications formelles, vérifiables et exécutables. Notre cadre applicatif porte sur les systèmes cyber-physiques, où des éléments informatiques sont conçus en interaction avec des entités physiques. Nous pouvons donner de nombreux exemples de systèmes cyber-physiques : un avion, une centrale électrique, un réseau ferroviaire, etc.

La conception de ces systèmes implique de décrire de manière précise, cohérente et complète toutes les caractéristiques fonctionnelles et comportementales du système cible. Un cahier des charges comprend généralement un ensemble d'exigences, décrivant précisément toutes les propriétés du système à réaliser. De plus, il rassemble différents points de vues, en mettant notamment en évidence les interactions des entités concernées. C'est une tâche complexe et source d'erreurs. L'outillage informatique qui pourrait être proposé implique le traitement de ces exigences et nécessite préalablement une représentation formelle de ces dernières. Pour l'obtenir, nous pouvons imposer au concepteur des contraintes sur la rédaction du cahier des charges, par exemple en proposant une interface formelle spécifique. Cette orientation représente une difficulté supplémentaire pour le concepteur, qui s'ajoute à celle de la problématique technique auquel il doit répondre. Nous proposons ici une autre approche qui vise à construire automatiquement les représentations formelles à partir d'énoncés exprimés en langue naturelle.

Nous nous intéressons plus particulièrement à l'aspect comportemental des systèmes étudiés, spécifié avec un ensemble d'exigences. Celles-ci décrivent des enchaînements d'action, et contiennent généralement des contraintes temporelles et des modalités déontiques. Les contraintes temporelles précisent la portée d'un évènement ou d'une propriété, tandis que les modalités déontiques expriment des obligations, des interdictions ou des possibilités. Par exemple, les énoncés suivants caractérisent l'état attendu d'une porte d'un système quelconque :

— La porte doit s'ouvrir lorsqu'un ticket est présenté. (obligation)
— L'ouverture de la porte est interdite après 21 heures. (interdiction)
— La porte peut s'ouvrir manuellement. (possibilité)

La logique de notre démarche est d'aboutir à la proposition de logiciels intégrés aux outils existants, favorisant ainsi une bonne acceptation des utilisateurs. Cet objectif tend à exclure l'utilisation d'un langage de domaine spécifique (DSL) pour privilégier la langue naturelle.

En traitement automatique des langues, les approches à dominante statistique sont généralement distinguées de celles à dominante symbolique. Nous proposons d'utiliser un formalisme de haut-niveau, à dominante symbolique : les grammaires catégorielles (Steedman, 2000; Moot & Retoré, 2012). Notre volonté est de proposer une représentation la plus juste et fine possible des exigences exprimées. Les grammaires catégorielles permettent une analyse compositionnelle, un développement incrémental et une interface aisée avec la sémantique. Ces points sont pertinents et importants si nous voulons obtenir une représentation qui capture le sens logique des phrases, avec un résultat manipulable et exploitable pour la vérification d'exigences.

Un cahier des charges peut se ramener à un ensemble de propriétés se rapportant aux composants d'une architecture. Partant de cet ensemble, nous voulons aboutir à une représentation du système. Sur le plan opératoire, nous pouvons décomposer notre problématique en deux phases (figure 1).

La première phase est l'analyse formelle des exigences, qui doit permettre d'aboutir à leur représentation sémantique. Une exigence est un énoncé en une phrase, parfois quelques phrases, donnant une spécification comportementale sur un point précis. La phrase "Si un ticket est inséré, la porte peut s'ouvrir" constitue un exemple d'exigence sur une porte dont l'ouverture est conditionnelle. Nous proposons de construire les représentations sémantiques en nous appuyant sur un formalisme grammatical.

La seconde phase consiste en la composition de ces exigences pour obtenir une représentation du comportement du système. Deux exigences ne se composeront pas nécessairement de la même manière ; il est donc nécessaire que les propriétés algébriques du modèle utilisé permettent l'usage de

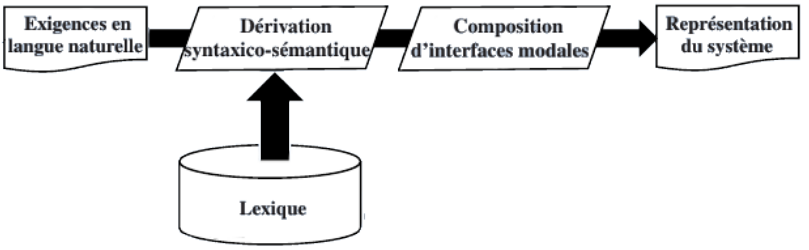


FIGURE 1 – Vue d'ensemble de la méthodologie proposée

différentes opérations de composition.

Cet article est un document d'analyse et de positionnement, présentant différents concepts qui pourraient être combinées. La section 2 introduit un formalisme pour l'analyse sémantique d'une exigence, les grammaires catégorielles combinatoires (Steedman, 2000), tandis que la représentation du comportement des systèmes techniques est étudiée, section 3, avec l'introduction des spécifications modales (Larsen & Thomsen, 1988; Raclet, 2007). Quelques travaux préliminaires sont présentés dans la section 4. Les perspectives offertes par l'approche proposée sont finalement discutées dans la section 5.

2 Analyse d'exigence

L'approche que nous proposons vise à construire une représentation logique d'un énoncé en "capturant" finement sa sémantique. Suivant les travaux de Montague (Montague, 1974), nous considérons le sens d'une phrase comme lié à sa construction syntaxique. Nous partons d'une décomposition de l'énoncé à l'aide d'un formalisme grammatical, pour ainsi définir le comportement syntaxique de l'énoncé cible. L'idée est ensuite de relier cette décomposition à une représentation formelle.

Comme indiqué en introduction, nous proposons d'utiliser les grammaires catégorielles, un formalisme de haut-niveau pour l'analyse de textes. Les grammaires catégorielles ont pris forme dans les années 50 et 60, à partir des travaux de Adjukiewicz (Ajdukiewicz, 1935), Bar-Hillel (Bar-Hillel, 1953) et Lambek (Lambek, 1958). Elles reposent sur la notion de catégories primitives et de catégories dérivées (on parle aussi de types). Elles offrent la possibilité de décomposer une phrase en catégories, les catégories composées mettant en évidence les liens syntaxiques du langage ciblé. Les grammaires catégorielles intègrent également des règles de dérivation qui permettent de combiner les éléments d'une phrase.

Du point de vue de la théorie grammaticale, il est important de considérer l'expressivité des formalismes étudiés. Pour les grammaires catégorielles classiques, elle est équivalente aux grammaires non-contextuelles (Bar-Hillel et al, 1963). Plusieurs extensions ont donc été proposées pour dépasser cette contrainte, dont la grammaire catégorielle combinatoire (CCG), introduite par Mark Steedman en 1989 (Steedman, 2000). Ce formalisme utilise un ensemble d'opérateurs pour saisir d'autres aspects linguistiques. Dans cette grammaire, la sémantique des mots est guidée tout au long de l'arbre de dérivation en combinant les sémantiques des autres mots jusqu'à produire la sémantique complète de la phrase, à la Montague (Montague, 1974). Elle est particulièrement intéressante dans notre cas

pour ses liens avec le lambda-calcul, qui pourrait être une représentation formelle intermédiaire avant d'aboutir à d'autres représentations logiques.

Concrètement, les grammaires CCG permettent de relier la syntaxe des énoncés étudiés à leur représentation sémantique sous-jacente. Elles se définissent avec un ensemble de catégories basiques, un ensemble de catégories dérivées, des règles de calcul syntaxiques et un lexique de mots.

Les catégories primitives correspondent aux catégories syntaxiques habituellement utilisées dans les grammaires traditionnelles. Cet ensemble de catégories primitives n'est pas universel mais fait partie de la définition de la grammaire. Il est possible de se restreindre à deux catégories : *S* pour la phrase ("sentence" en anglais), *N* pour les noms communs et les noms propres. Il est également possible d'étendre cet ensemble en ajoutant des catégories, par exemple pour les adjectifs ou les pronoms. De même, la grammaire peut-être raffiner avec des sous-catégories, comme sur la figure 2.

Les catégories dérivées sont construites à partir des catégories primitives. Elles peuvent se définir récursivement avec les règles syntaxiques : soit X et Y deux catégories, alors X/Y et $X\backslash Y$ sont également des catégories [1].

Les règles syntaxiques permettent de produire l'analyse sémantique en combinant les entrées lexicales. Les grammaires catégorielles classiques proposent deux règles de calcul, également présentes avec les CCG : l'application en avant ($>$) et l'application en arrière ($<$). Chacune de ces règles permet de former une nouvelle catégorie en combinant une catégorie de type X/Y ou $X\backslash Y$ avec une catégorie de type Y. Formellement, elles peuvent se définir comme suit :

— $X/Y, Y \rightarrow X$ ($>$)
— $Y, X\backslash Y \rightarrow X$ ($<$)

Les CCG enrichissent les grammaires catégorielles classiques avec de nouvelles règles de calcul. Nous définissons ici les règles de composition harmoniques, en avant ($B>$) et en arrière ($B<$) :

— $X/Y, Y/Z \rightarrow X/Z$ ($B>$)
— $Y\backslash X, Z\backslash Y \rightarrow Z\backslash X$ ($B<$)

D'autres règles sont également proposées : des règles de compositions croisées, des règles de composition mixtes et des règles de changement de types. L'ensemble de ces règles offre différentes façons de combiner les catégories, et introduisent un certain degré d'associativité. Elle permettent ainsi de traiter des aspects linguistiques plus riches.

Finalement, le lexique permet de définir la grammaire qui correspond à un langage donné. Il est constitué d'un ensemble d'entrées lexicales associant un mot et un ensemble de types, auxquelles il est possible d'adjoindre une représentation sémantique. Cette dernière peut être donnée sous la forme d'expressions en lambda-calcul [2]. Par exemple, le mot "gate" pourra être associé à la catégorie N, tandis que la modalité "may" serait associée à la catégorie $(S\backslash NP)/(S\backslash NP)$. Ces deux entrées

1. Il existe plusieurs systèmes de notation. Dans la notation de Lambek, X/Y se lit X sur Y et signifie qu'un mot (ou groupe de mots) de catégorie Y est attendu à droite pour former la catégorie X. Par symétrie, $X\backslash Y$ se lit X sous Y et signifie qu'un mot ou groupe de mots de catégorie X est attendu à gauche pour former la catégorie Y. Dans la notation de Steedman, X/Y et $X\backslash Y$ définissent une catégorie avec un argument de type Y et un résultat de type X. Une barre oblique (/) indique que l'argument doit apparaître à droite, une barre oblique inversée (\) indique que l'argument doit apparaître à gauche. C'est cette dernière notation que nous utilisons dans cet article.

2. Le lambda-calcul est un formalisme qui permet de définir et caractériser les fonctions mathématiques. En particulier, ce formalisme offre la possibilité de modéliser des expressions fonctionnelles et leur évaluation en manipulant des expressions, appelées lambda-termes, dans lesquelles une abstraction désigne une définition de fonction. Ce formalisme a été proposé par Alonzo Church dans les années 1930.

pourraient être rattachées respectivement aux formules $gate$ et "$\lambda x.(x, may)$".

La figure 2 montre un exemple d'analyse avec le parseur EasyCCG [3] (Lewis & Steedman, 2014) pour la phrase "After a ticket is inserted, the gate may open". L'analyse syntaxique doit permettre de réduire la séquence des catégories associées aux mots à la catégorie principale de la phrase S, ce que nous observons sur cette exemple. Les éléments entre crochets correspondent à des étiquettes grammaticales permettant d'affiner l'analyse syntaxique.

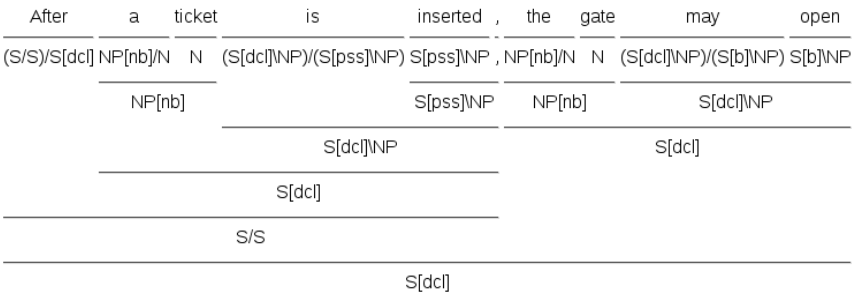


FIGURE 2 – Dérivation CCG de la phrase "After a ticket is inserted, the gate may open" construite avec le parser EasyCCG

Les dérivations obtenues avec une analyse CCG peuvent être reliées à une représentation sémantique, en s'appuyant sur les associations définies dans le lexique. Des algorithmes d'analyse CCG en temps polynomial existent (Vijay-Shanker & Weir, 1993). La mise en oeuvre de ce modèle s'est traduite dans le développement de plusieurs outils, dont l'outil Boxer/C&C (Bos, 2015) utilisé dans les travaux préliminaires présentés en section 4.

3 Représentation du comportement d'un système

Les systèmes cyber-physiques sont généralement complexes. Ils résultent de l'assemblage de plusieurs composants, conçus par des équipes travaillant indépendamment. Ces équipes s'accordent sur les attentes pour chaque composant et leur intégration commune en s'appuyant sur les spécifications techniques issues du cahier des charges. Pour vérifier la consistance et la complétude des exigences d'un cahier des charges, il est nécessaire de disposer d'un formalisme de spécification. Ce formalisme doit offrir un bon compromis en termes d'expressivité, de propriétés algébriques et de complexité algorithmique.

Les automates ne permettent pas d'exprimer la variabilité des points de vues et des composants. Les automates d'interfaces (de Alfaro & Henzinger, 2001) n'ont pas l'opération de conjonction, tandis que les logiques temporelles LTL et CTL (Clarke & Emerson, 1981; Clarke *et al.*, 1986) n'ont pas la composition parallèle. Le μ-calcul modal (Arnold & Niwiński, 2001; Arnold *et al.*, 2003) possède les bonnes propriétés algébriques, avec un niveau d'expressivité élevé mais une complexité de calcul également importante ($O(2^{2^N})$).

3. EasyCCG est un parseur CCG open-source. Il est disponible à l'adresse suivante : http ://homepages.inf.ed.ac.uk/s1049478/easyccg.html.

Les spécifications modales (Raclet, 2007) représentent le compromis recherché. En effet, correspondant aux systèmes de transitions modaux déterministes (Larsen & Thomsen, 1988), ce formalisme intègre les opérations de conjonction, de composition et de quotient avec une complexité algorithmique polynomiale. Il est donc adapté pour la représentation d'interfaces des différents composants d'un système. Certes, les spécifications modales sont strictement moins expressives que le μ-calcul modal . Cette limitation est néanmoins acceptable pour formaliser les propriétés de sureté des cahiers des charges de systèmes cyber-physiques.

Ce formalisme permet de représenter des automates dont les transitions sont typées avec les modalités *must* et *may*, permettant de définir un ensemble de modèles. De manière informelle, une transition de type *must* (*obligatoire*) doit être présente dans chaque composant qui implémente la spécification modale, alors qu'une transition de type *may* (*facultative*) peut ne pas l'être.

Les spécifications modales sont des systèmes de transitions modaux déterministes (Raclet, 2007). Les systèmes de transitions modaux ont été introduits par Larsen et Thomsen dans les années 80 (Larsen & Thomsen, 1988; Jonsson & Larsen, 1991) pour étudier le raffinement d'actions, opération qui consiste à remplacer des transitions pour relier les descriptions d'un système à différents niveaux d'abstraction.

Nous reprenons la définition formelle de J.-B. Raclet (Raclet, 2007). Une spécification modale est un tuple $S = (Q, q_0, \Sigma, \Delta^m, \Delta^M)$. Q est un ensemble fini d'états. q_0 est l'unique état initial. Σ est un ensemble fini d'actions. $\Delta^m \subseteq Q \times \Sigma \times Q$ est l'ensemble déterministe des transitions facultatives ; $\Delta^M \subseteq Q \times \Sigma \times Q$ est l'ensemble déterministe des transitions obligatoires. Enfin, la condition de consistance suivante doit être respectée : $\Delta^M \subseteq \Delta^m$ et $q_0 \in Q$. Cette condition signifie que chaque transition obligatoire (appartenant à l'ensemble Δ^M) est également une transition facultative (appartenant à l'ensemble Δ^m).

Le modèle d'une spécification modale est défini à partir de la relation de satisfaction, notée $\models$. Soit $S = (Q, q_0, \Sigma, \Delta^m, \Delta^M)$ une spécification modale. Soit $M = (R, r_0, \Sigma, \Gamma \subseteq R \times \Sigma \times R)$ avec Γ déterministe. M est un modèle de S, noté $M \models S$, si, et seulement si, il existe $\rho \subseteq R \times Q$ tel que :

1. $(r_0, q_0) \in \rho$
2. $\forall (r, q) \in \rho, \forall \sigma \in \Sigma$
   (a) $\forall r' \in R, (r, \sigma, r') \in \Gamma \Rightarrow \exists q' \in Q$ tel que $(q, \sigma, q') \in \Delta^m$ et $(r', q') \in \rho$
   (b) $\forall q' \in Q, (q, \sigma, q') \in \Delta^M \Rightarrow \exists r' \in R$ tel que $(r, \sigma, r') \in \Gamma$

Il est également possible de définir une représentation graphique de spécifications modales. Ces représentations produisent des exemples visuels sous la forme de graphes avec des arcs continus ou pointillés. La présence d'un arc continu signifie que la transition est obligatoire ("must"), la présence d'un arc en pointillé que la transition est facultative ("may") et l'absence d'arc indique que la transition est interdite dans toute réalisation.

La figure 3 montre une spécification modale (à gauche) et deux modèles possibles de cette spécification (à droite), où a et b sont deux actions (ou évènements).

Les spécifications modales sont dotées de propriétés algébriques particulièrement intéressantes. Ces propriétés sont détaillées dans (Raclet *et al.*, 2010). On y trouvera en particulier les définitions formelles (et preuves associées) pour la composition ($S_1 \otimes S_2$), la conjonction ($S_1 \wedge S_2$) et le quotient (S_1/S_2) de spécifications modales (avec S_1 et S_2 deux spécifications modales).

Nous précisons ici quelques points importants : la notion de raffinement, et les caractéristiques algébriques des opérations évoquées. Une spécification modale S_1 est le raffinement d'une spéci-

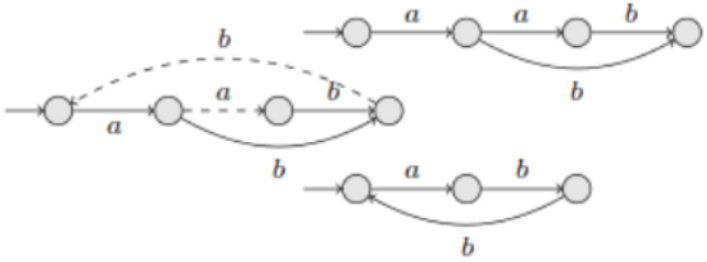


FIGURE 3 – Une spécification modale et deux modèles possibles

fication S_2 si, et seulement si, les modèles de S_1 sont aussi des modèles de S_2. Formellement, le raffinement, noté $\preceq$, se définit comme suit : $S_1 \preceq S_2$ si, et seulement si, pour tout modèle M, nous avons $M \models S_1 \Rightarrow M \models S_2$. Nous pouvons dès lors donner les caractéristiques algébriques des opérations de composition, conjonction et quotient, en nous appuyant sur le raffinement :

— $S_1 \otimes S_2 = min\{S \mid \forall M_1 \models S_1, \forall M_2 \models S_2, M_1 \times M_2 \models S\}$
— $S_1 \wedge S_2 = min\{S \mid \forall M, M \models S_1 et M \models S_2 \Rightarrow M \models S\}$
— $S_1/S_2 = max\{S \mid S_2 \otimes S \leq S_1\}$

Les spécifications modales apparaissent comme un formalisme pertinent pour la représentation des exigences d'un système réactif. Elles peuvent prendre la forme d'automates qu'il est possible d'exécuter pour valider un cahier des charges. De plus, elles sont implémentables avec une complexité raisonnable, et permettent de représenter les modalités déontiques et temporelles présentes dans la définition d'un système.

4 Travaux préliminaires

Nous proposons de montrer une mise en oeuvre possible des concepts exposés dans les sections précédentes sur un cas d'étude. L'objectif visé est la construction de la spécification modale d'un système dont le comportement est décrit en langue naturelle. Pour atteindre ce but, nous présentons une méthode de traitement sur un ensemble d'exigences. Celle-ci doit permettre de relier les dérivations syntaxiques de ces exigences à des représentations sémantiques pertinentes.

L'exemple présenté ici repose sur des expérimentations préliminaires (Boudaoud, 2016) portant sur un système représentant un garage de voitures (Benveniste *et al.*, 2018). La figure 4 propose une vue d'ensemble du document de spécifications, composé de plusieurs sections (*gate*, *payment*, *supervisor*).

Pour ces travaux préliminaires, les données utilisées proviennent d'un cas d'étude théorique. De plus, les exigences exprimées en plusieurs phrases, avec des formulations imposant la prise en compte du contexte, ont été mises de côté. Chaque exigence est donc définie en une phrase et sans pronom,

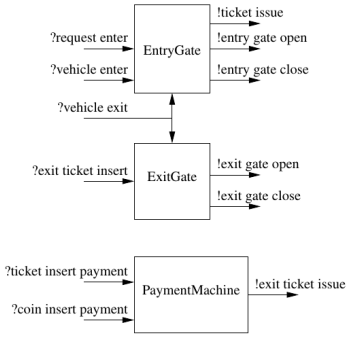


FIGURE 4 – Architecture du système "The Parking Garage" (Benveniste *et al.*, 2018)

chaque entité devant être explicitement désignée dans l'exigence [4].

Les représentations sémantiques sont obtenues à partir de dérivations CCG, comme définies section 2. Concrètement, chaque élément lexical est associé à une catégorie lexicale et une représentation sémantique exprimée en lambda-calcul. Il est également possible de donner plusieurs représentations pour un même élément lexical (dans le cas où l'élément lexical serait associé à plusieurs catégories syntaxiques). Le traitement implique ensuite de déterminer les fonctions et les arguments, et d'appliquer récursivement les fonctions sur les arguments jusqu'à obtenir la représentation sémantique complète de l'exigence ciblée.

Le passage des formules en lambda-calcul vers les spécifications modales comporte plusieurs difficultés, dont la mise en relation des termes de l'énoncé (source) aux états et actions de la spécification (cible). Au niveau de l'implémentation, un module spécifique pourrait permettre de gérer cette transformation. Ce module doit également intégrer la résolution des références aux concepts, par exemple en s'appuyant sur une ontologie du domaine d'étude.

Un automate peut être spécifié sous la forme d'une expression régulière, dont la définition peut prendre en compte les opérateurs algébriques tels que la concaténation, le choix, la répétition ou le mot vide. Des règles de transformation associant une expression régulière à chaque terme en langue naturelle peuvent être proposées en s'appuyant sur cette définition. L'application de ces règles permet de décrire les propriétés d'états en expression régulière. Sur le même principe, nous pouvons associer des transitions aux termes en langue naturelle définissant des modalités. La combinaison de ces relations permet de construire la représentation des états et transitions sous la forme d'une spécification modale, à partir de l'arbre de dérivation CCG.

Ce cas d'étude a été étudié en détail, du point de vue *"conception de système"*, dans (Benveniste *et al.*, 2018). Il a été mis en oeuvre avec l'outil MICA (Caillaud, 2011), une bibliothèque qui permet de définir des systèmes complexes en utilisant une syntaxe du langage O'Caml. Dans cette mise en oeuvre, la formalisation des exigences a été réalisée manuellement (les exigences ont été traduites dans la syntaxe exigée par l'outil Mica). L'objectif serait d'enrichir cet outil pour pouvoir traiter

4. Par exemple, l'exigence "Quand la porte est fermée, elle ne peut pas être ouverte tant qu'aucun ticket n'est inséré." n'a pas été traitée sous cette forme car la prise en compte du pronom "elle" pose quelques difficultés. Elle peut l'être en revanche sous la forme suivante : ""Quand la porte est fermée, la porte ne peut pas être ouverte tant qu'aucun ticket n'est inséré". Néanmoins, des travaux complémentaires, en s'appuyant sur les DRT, devraient permettre de lever cette restriction.

directement les exigences exprimées en langue naturelle. Dans le tableau 1, nous reprenons quelques exigences portant sur la section "gate".

Id.	*Exigence*
1	Entry gate must open once a ticket has been issued.
2	When a ticket has been issued, entry gate must open.
3	When a ticket has been issued, entry gate opening is obligatory.
4	Afer an exit ticket is inserted, exit gate must open.

TABLE 1 – Quelques exigences du système "The parking garage"

Il est à noter que les exigences contiennent plusieurs caractéristiques linguistiques qui doivent être prises en compte. Elles peuvent décrire des enchaînements d'évènements, avec des modalités temporelles. Elles peuvent également contenir des éléments qui font référence à des entités de l'environnement du système (par exemple un véhicule), ou des informations implicites. Les spécifications contiennent aussi des modalités d'obligation et d'interdiction. L'extraction des informations nécessaires pour le passage vers une spécification modale implique une bonne évaluation de ces différents aspects.

L'outil Boxer, développé par Johan Bos (Bos, 2015), a été utilisé pour traiter les exigences et construire des représentations sémantiques. Boxer/C&C est un analyseur sémantique de textes en anglais. Il s'appuie sur la grammaire catégorielle combinatoire et la théorie de représentation du discours (Kamp, 1981), et permet de construire des représentations sémantiques sous différentes formes. La figure 5 donne une représentation de l'exigence 4 du tableau 1.

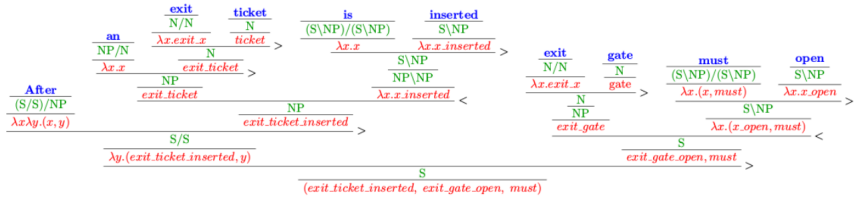


FIGURE 5 – Représentation graphique de l'analyse sémantique pour l'exigence 4

Le lien entre Boxer et Mica n'a pas encore été réalisé. Sur le principe, il s'agit de générer du code O'Caml à partir des représentations formelles obtenues. La figure 6 donne une vue du résultat que nous pourrions obtenir sous la forme d'une spécification modale en prenant en compte les exigences de la section "Gate". Cette représentation intègre l'analyse de l'exigence 4 (figure 5). On retrouve cette propriété au niveau des états 0 et 1.

Même s'il est incomplet, le résultat obtenu montre la pertinence de l'approche. Il serait souhaitable de développer un outil complet permettant l'implémentation d'interfaces modales pour des composants spécifiés en langue naturelle. Sa réalisation pourrait s'appuyer sur les outils présentés, Boxer pour la dérivation CCG, et Mica pour les interfaces modales, ou d'autres outils aux caractéristiques similaires. Pour valider cette démarche, il conviendrait également d'évaluer l'approche sur l'ensemble des exigences du cas d'étude, ainsi que sur des données issues d'un cas réel.

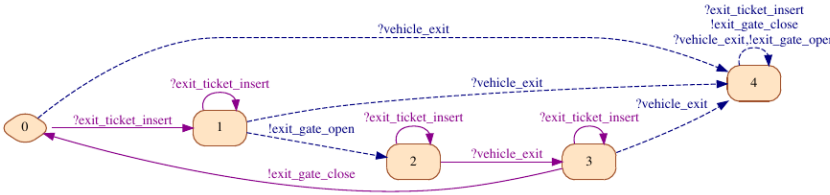


FIGURE 6 – Représentation graphique de la spécification modale pour la section "Gate" du cas d'étude "The Parking garage" (Benveniste *et al.*, 2018)

5 Discussion et perspectives

L'analyse des besoins est la première étape du cycle de développement d'un système. L'élaboration du cahier des charges intervient donc très tôt dans ce cycle ; le coût d'une erreur non détectée à ce stade est important. Le passage des spécifications en langage naturel vers des spécifications formelles vérifiables suscite un intérêt évident. Plusieurs approches ont déjà été proposées. Elles utilisent généralement une représentation intermédiaire pour la formalisation de spécifications en langue naturelle, par exemple une modélisation des connaissances par une ontologie (Sadoun *et al.*, 2013), des patrons de spécifications de propriétés temporelles (Dwyer *et al.*, 1999; Konrad & Cheng, 2005), un langage naturel contrôlé (Fuchs & Schwitter, 1995).

A contrario, l'approche que nous proposons vise à construire automatiquement les représentations formelles à partir d'énoncés exprimés en langue naturelle. Même si l'écriture des énoncés devra être d'abord restreinte à un langage contrôlé, notre objectif est de permettre de dépasser cette contrainte et rejoindre dans une certaine mesure la flexibilité du langage naturel.

Pour atteindre notre objectif, plusieurs étapes sont nécessaires. La première étape consiste en la dérivation d'une analyse syntaxique de l'énoncé, pour aboutir à un arbre syntaxique. La deuxième étape vise à relier l'analyse syntaxique à une représentation formelle intermédiaire, par exemple le lambda-calcul. Nous pensons possible de passer de cette représentation à une autre forme. Nous avons notamment montré l'intérêt des spécifications modales pour la vérification d'un cahier des charges.

Concernant le passage de la langue naturelle vers une représentation formelle, nous avons évoqué une première piste, à savoir l'utilisation des CCG, couplée aux DRT si nous voulons traiter les pronoms. D'autres options sont également envisageables et pourraient être étudiées. Nous pensons notamment aux travaux autour du "Grammatical Framework" [5], également fondé sur les grammaires catégorielles et la théorie des types. Nous songeons aussi aux Grammaires Catégorielles Abstraites (de Groote, 2001), formalisme qui généralise les grammaires catégorielles en traitant directement les catégories et le lambda-calcul.

Nous avons évoqué la théorie de représentation du discours (DRT), proposé par Hans Kamp en 1983 (Kamp, 1981). Ce modèle permet de construire une représentation d'un texte en examinant le contenu sémantique dépendant du contexte. Il permet ainsi de traiter certaines difficultés linguistiques, en particulier les mots grammaticaux se substituant à un élément donné (pronoms). A la base de la DRT, il y a l'idée que l'interprétation d'un discours s'appuie sur une représentation sémantique, chaque

5. Le système GF, qui combine langage source et langage cible, a été utilisé dans différents projets. Il est accessible à l'adresse suivante : https ://www.grammaticalframework.org/

phrase enrichissant cette représentation. Cette représentation est appelée "structure de représentation du discours" (DRS). La DRT intègre également une procédure de construction, pour mettre à jour la représentation à chaque nouvelle phrase, et un ensemble de modèle sémantique, pour évaluer les DRS. Ce formalisme pourrait être exploité conjointement aux CCG pour augmenter la couverture des phénomènes linguistiques.

La validation de la méthodologie proposée passe par une mise en oeuvre concrète. Une première étape serait de développer un prototype visant l'implémentation de spécifications modales à partir d'exigences exprimées dans un langage contraint, le plus proche possible de la langue naturelle. Dans un second temps, il s'agirait de lever les contraintes sur le texte pour se rapprocher au maximum de la flexibilité de la langue naturelle.

De nombreuses questions se posent alors, en particulier pour le traitement des énoncés ambigus. L'interaction entre le programme et l'utilisateur est une première réponse. Nous pourrions imaginer que le système propose plusieurs variantes pour l'énoncé. Cependant, cela demanderait un pré-requis important pour l'usage de l'outil, à savoir la maîtrise du langage formel utilisé. Cette solution ne serait pas satisfaisante sous cette forme. Il serait donc intéressant de travailler sur le cheminement inverse, c'est-à-dire le passage d'une représentation formelle vers un énoncé en langue naturelle intelligible.

Nous pourrions d'ailleurs pousser la réflexion sur le développement de l'interaction. La théorie de la représentation du discours pourrait être une source d'inspiration. L'idée serait la suivante : l'analyse de l'énoncé permet de construire une première représentation avec un certain niveau d'ambiguïté ; les interactions du système avec l'utilisateur conduisent à la mise à jour de cette représentation, qui est validée lorsqu'il n'y a plus d'ambiguïté.

L'utilisation d'une ontologie pour représenter les concepts du domaine est une autre piste, alternative ou complémentaire. Celle-ci nous semble particulièrement adaptée pour le traitement des ambiguïtés dans les groupes nominaux. En effet, un groupe nominal, éventuellement complexe, doit correspondre à un concept de l'ontologie, et ainsi être vu comme atomique. Les synonymes pourraient être traités de manière similaire, un même concept pouvant correspondre à plusieurs groupes nominaux.

Ces deux axes pourraient être complémentaires. La construction d'une ontologie est cependant une tâche délicate qui demande un investissement important. Travailler sur une ontologie incomplète pourrait être une nécessité. Il serait instructif d'examiner les possibilités d'apprentissage des concepts par le système, et de la mise à jour interactive de l'ontologie de référence. Par exemple, le cahier des charges pourrait contenir des définitions qui pourraient être prises en compte pour mettre à jour l'ontologie. De même, les différents échanges du système avec l'utilisateur pourraient être pris en compte, avec un certain degré de confiance sur les concepts enregistrés. Par ricochet, de nouvelles problématiques apparaissent : comment quantifier ce degré de confiance ?

Et même au delà, l'un des objectifs étant de vérifier la cohérence des exigences, une réflexion devra être menée sur la mise en évidence des incohérences. Autrement dit, quel résultat proposer lorsque le cahier des charges est inconsistant ? Il pourrait alors être utile d'offrir au concepteur la possibilité de simuler un cahier des charges, traduit dans un formalisme tel que les spécifications modales, et de modifier les spécifications de manière interactive, en cours de simulation.

Un autre axe de travail consisterait à étendre les interfaces modales à des formalismes plus expressifs, et à tenter de prendre en compte les propriétés numériques des systèmes cyber-physiques. Les propriétés numériques permettent de définir des enveloppes de trajectoires pour les systèmes cyber-physiques, prenant généralement la forme d'équations ou d'inclusions différentielles qui caractérisent la dynamique en temps continu du système.

Dans cet article, nous avons mis en lumière différents formalismes qui nous semblent utiles pour répondre à notre problématique. Notre présentation est loin d'être exhaustive. Il existe en effet de nombreux formalismes grammaticaux, sans réel consensus. Il serait intéressant de comparer ces formalismes, ce qui pourrait nous conduire à orienter nos travaux dans de nouvelles directions.

Références

AJDUKIEWICZ K. (1935). Die syntaktische konnexität. *Studia Philosophica*, **1**, 1–27. Traduction française de (Gan-Krzywoszyńska, 2007).

ARNOLD A. & NIWIŃSKI D. (2001). *Rudiments of [mu]-calculus*. Studies in Logic and the Found. Elsevier.

ARNOLD A., VINCENT A. & WALUKIEWICZ I. (2003). Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, **303**(1), 7–34. Logic and Complexity in Computer Science.

BAR-HILLEL Y. (1953). A quasi-arithmetical notation for syntactic description. *Language*, **29**, 47–58.

BENVENISTE A., CAILLAUD B., NICKOVIC D., PASSERONE R., RACLET J.-B., REINKEMEIER P., SANGIOVANNI-VINCENTELLI A., DAMM W., HENZINGER T. A. & LARSEN K. G. (2018). Contracts for system design. *Foundations and Trends® in Electronic Design Automation*, **12**(2-3), 124–400.

BOS J. (2015). Open-domain semantic parsing with boxer. *Proceedings of the 20th Nordic Conference of Computational Linguistics*, p. 301–304.

BOUDAOUD S. R. (2016). Spécification d'exigences en langue naturelle, avec automates et logique. Master's thesis, Université de Rennes 1. Private communication.

CAILLAUD B. (2011). Mica : A Modal Interface Compositional Analysis Library.

CLARKE E. M. & EMERSON E. A. (1981). Synthesis of synchronization skeletons for branching time temporal logic. In *Logic of programs : Workshop*, volume 131, p. 244–263.

CLARKE E. M., EMERSON E. A. & SISTLA A. P. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, **8**(2), 244–263.

DE ALFARO L. & HENZINGER T. A. (2001). Interface automata. *SIGSOFT Softw. Eng. Notes*, **26**(5), 109–120.

DE GROOTE P. (2001). Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, p. 252–259 : Association for Computational Linguistics.

DWYER M. B., AVRUNIN G. S. & CORBETT J. C. (1999). Patterns in property specifications for finite-state verification. In *Proceedings of the 21st International Conference on Software Engineering*, ICSE '99, p. 411–420, New York, NY, USA : ACM.

FUCHS N. E. & SCHWITTER R. (1995). Specifying logic programs in controlled natural language. *CoRR*, **abs/cmp-lg/9507009**.

GAN-KRZYWOSZYŃSKA K. (2007). La connexion syntaxique. *Philosophia Scientiæ*, **11-2**, 97–120.

JONSSON B. & LARSEN K. G. (1991). Specification and refinement of probabilistic processes. In *[1991] Proceedings Sixth Annual IEEE Symposium on Logic in Computer Science*, p. 266–277.

KAMP H. (1981). A theory of truth and semantic representation. In J. A. G. GROENENDIJK, T. M. V. JANSSEN & M. B. J. STOKHOF, Eds., *Formal Methods in the Study of Language*, volume 1, p. 277–322. Amsterdam : Mathematisch Centrum.

KONRAD S. & CHENG B. H. C. (2005). Real-time specification patterns. In *Proceedings of the 27th International Conference on Software Engineering*, ICSE '05, p. 372–381, New York, NY, USA : ACM.

LAMBEK J. (1958). The mathematics of sentence structure. *American Mathematical Monthly*, **154-170**.

LARSEN K. G. & THOMSEN B. (1988). A modal process logic. In *Proceedings. Third Annual Symposium on Logic in Computer Science*, p. 203–210.

LEWIS M. & STEEDMAN M. (2014). A* ccg parsing with a supertag-factored model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processin*.

MONTAGUE R. (1974). *Formal Philosophy : Selected Papers of Richard Montague*. A Yale Paperbound. Yale University Press.

MOOT R. & RETORÉ C. (2012). *The Logic of Categorial Grammars : A Deductive Account of Natural Language Syntax and Semantics*. FoLLI-LNCS. Springer.

RACLET J. (2007). *Quotient de spécifications pour la réutilisation de composants*. PhD thesis, Université de Rennes 1.

RACLET J., BADOUEL E., BENVENISTE A., CAILLAUD B., LEGAY A. & PASSERONE R. (2010). A modal interface theory for component-based design. *Fundamenta Informaticae*, **108**(1-2), 119–149.

SADOUN D., DUBOIS C., GHAMRI-DOUDANE Y. & GRAU B. (2013). From natural language requirements to formal specification using an ontology. In *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI 2013)*, p. 755–760, Herndon, VA, United States.

STEEDMAN M. (2000). *The syntactic process*. MIT Press.

VIJAY-SHANKER K. & WEIR D. J. (1993). Parsing some constrained grammar formalisms. *Comput. Linguist.*, **19**(4), 591–636.

Résumé automatique guidé de textes : État de l'art et perspectives

Salima Lamsiyah[1] Saïd Ouatik El Alaoui[1] Bernard Espinasse[2]
(1) LIM, Faculté des Sciences Dhar El Mahraz, Université Sidi Mohamed Ben Abdellah, Fès, Maroc
(2) LSIS-UMR CNRS, Aix-Marseille Université, Marseille, France
Salima.lamsiyah@usmba.ac.ma, said.elalaouiouatik@usmba.ac.ma,
bernard.espinasse@lis-lab.fr

RÉSUMÉ

Les systèmes de résumé automatique de textes (SRAT) consistent à produire une représentation condensée et pertinente à partir d'un ou de plusieurs documents textuels. La majorité des SRAT sont basés sur des approches extractives. La tendance actuelle consiste à s'orienter vers les approches abstractives. Dans ce contexte, le résumé guidé défini par la campagne d'évaluation internationale TAC (Text Analysis Conference) en 2010, vise à encourager la recherche sur ce type d'approche, en se basant sur des techniques d'analyse en profondeur de textes. Dans ce papier, nous nous penchons sur le résumé automatique guidé de textes. Dans un premier temps, nous définissons les différentes caractéristiques et contraintes liées à cette tâche. Ensuite, nous dressons un état de l'art des principaux systèmes existants en mettant l'accent sur les travaux les plus récents, et en les classifiant selon les approches adoptées, les techniques utilisées, et leurs évaluations sur des corpus de références. Enfin, nous proposons les grandes étapes d'une méthode spécifique devant permettre le développement d'un nouveau type de systèmes de résumé guidé.

ABSTRACT

Guided Summarization : State-of-the-art and perspectives

Automatic text summarization (ATS) aims to produce from one or more texts, a summary that represents the most relevant information included in the original textual sources. Most existing ATS are mainly the extraction-based systems ; however, the trend today is to make a move toward abstraction-based systems. In 2010, the Text Analysis Conference (TAC) campaign defined the guided summarization task as a recent type of ATS, aims to encourage a deeper linguistic analysis of the source documents instead of relying only on classical extractive approaches. In this work, we provide an introduction to guided summarization task, by defining the different characteristics and constraints related to this task, and by reviewing the details of different guided summarization systems developed so far. We also classify these systems according to the adopted approaches, techniques used, and evaluations on reference corpus. Finally, we propose the main steps of a specific method that will allow the development of a new type of guided summary systems.

MOTS-CLÉS : Résumé automatique de textes, résumé guidé, approche extractive, approche abstractive, traitement automatique de la langue naturelle, extraction d'information.

KEYWORDS: Automatic text summarization, guided summarization, extraction-based approach, abstraction-based approach, natural language processing, information extraction.

1 Introduction

De nos jours, l'information textuelle en format numérique est abondante dans le web, elle représente une masse de 80% de l'information qui y circule. Dans la plupart des cas, cette immense quantité est non structurée : elle n'est pas sous forme de bases de données classiques, mais sous un format de texte libre nécessitant le besoin de concevoir et de développer des outils pertinents pour que l'utilisateur puisse accéder aux informations pertinentes.

Le résumé automatique de textes est un de ces outils, en condensant les textes de façon pertinente, le résumé automatique pourra être une solution efficace et éprouvée pour traiter cette masse grandissante d'informations.

Le résumé automatique de textes, apparu vers la fin des années 1950 (Luhn, 1958), a connu un fort renouveau ces dernières années. Produire automatiquement un résumé pertinent et de qualité nécessite de condenser le ou les documents originaux tout en minimisant la redondance, et en maximisant la cohérence et la cohésion. *La cohérence* est l'absence de contradictions et de la redondance dans l'enchainement des phrases d'un document. Pour toute partie d'un texte cohérent, il existe une fonction, une raison plausible à sa présence. *La cohésion* est un moyen de lier ensemble les différentes parties du texte, elle est assurée par l'utilisation de termes sémantiquement liés, co-références, ellipses et conjonctions. La cohésion se situe au niveau linguistique de la phrase, alors que la cohérence est située au niveau supérieur de la sémantique. De façon générale les systèmes de résumé automatique de textes (SRAT) sont confrontés à une difficulté majeure qui est l'absence d'un étalon-or unique (Gold standard) que les SART pourraient suivre, le résumé est guidé par une vague notion de l'importance des faits mentionnés dans les documents sources, ce qui est très subjectif et dépendant du contenu.

Une autre difficulté est liée à l'utilisation exclusive des approches extractives qui sont devenues le paradigme dominant du développement de SRAT. Bien que ces approches soient relativement simples, faciles à mettre en œuvre et efficaces en extraction des phrases pertinentes, elles sont loin de produire des résumés optimaux. En effet des problèmes de cohésion, la cohérence et de résolution d'anaphores empêchent les SRAT basés sur ce type d'approche de produire des bons résumés en termes du contenu et de la qualité linguistique. Les expériences HexTAC (Genest *et al.*, 2009) et d'autres études (Cremmins, 1993, 1996) ont montré que même le meilleur mécanisme contenu-sélection utilisé par les êtres humains est incapable de produire de bons résumés s'il se limite à assembler un ensemble de phrases prises hors de leurs contextes. En conséquence, la recherche en SRAT devrait s'orienter plus vers l'abstraction que vers l'extraction. L'une des thématiques récentes accompagnant cette tendance est le résumé guidé. Lancé dans la campagne internationale d'évaluation TAC'2010, le résumé guidé peut être considéré comme des résumés multi-documents dont les contenus sont déterminés par les besoins et les préférences des utilisateurs.

Le reste de cet article est organisé comme suit : la section 2 présente les différents types et approches de résumés automatiques de textes, la section 3 est dédiée au résumé guidé, sa définition, une architecture fonctionnelle générique, et les spécificités qui le distinguent des autres résumés. La section 4 présente brièvement plusieurs systèmes de résumé guidé et les compare selon différents critères. Dans la section 5 nous concluons et présentons différentes perspectives de recherche liées à l'amélioration des systèmes automatiques de résumé guidé.

2 Résumé automatique de textes

Bien que cet article s'intéresse au résumé guidé et à ses approches, nous présentons un aperçu des autres types de résumés, ainsi que les principales stratégies pour produire un résumé automatique.

2.1 Types de résumés automatiques de textes

Il existe plusieurs types de résumés de textes, du fait que l'on dispose de différents types et sources documentaires, et que le besoin en information diffère d'un utilisateur à un autre. Différentes taxonomies sont proposées pour les classifier (Sparck, 1998; Nenkova & McKeown, 2012). Nous présentons l'une des plus connues dans la littérature et qui classifie les résumés selon quatre critères (Lloret & Palomar, 2012) : l'entrée du SRAT, l'objectif, la sortie et la langue. En se basant sur le premier facteur, nous distinguons les SRAT *mono-documents* et les SRAT *multi-documents*. Les premiers produisent des résumés à partir d'un seul document alors que les deuxièmes génèrent des résumés pour un ensemble de documents et portant souvent sur une thématique bien précise.

Selon le critère objectif du SRAT, on distingue plusieurs types : le résumé indicatif, le résumé informatif, le résumé générique, le résumé orienté et le résumé de mise à jour. Le *résumé indicatif* a pour objectif d'aider le lecteur à agir sur sa décision à consulter ou pas un document, en lui indiquant les thématiques abordées et développées dans le document source, sans considérer les détails. Le *résumé informatif* a pour objectif principal de renseigner le lecteur sur les principales informations quantitatives et qualitatives, il est considéré comme une version abrégée, conservant l'organisation générale du document source. Le *résumé générique* résume le document sans prendre en compte les besoins en information des utilisateurs, par contre, le *résumé orienté* a pour objectif de ne résumer que les informations qui répondent à une requête de l'utilisateur. Le *résumé de mise à jour* se contente de fournir un résumé sous l'hypothèse que l'utilisateur a déjà des connaissances sur la thématique et qui n'a besoin que des nouveautés importantes, tout en évitant la redondance de l'information (Li *et al.*, 2009).

Selon le facteur de la langue, nous envisageons trois types des SRAT : monolingues, multilingues, et cross-lingues. Pour les *monolingues*, la source et le résumé sont écrits dans la même langue. Si le système peut traiter plusieurs langues, et produire des résumés dans la même langue que celle du document d'entrée, nous aurions un système *multilingue*. Si le résumé est en anglais, et que les documents originaux sont dans une autre langue, le SRAT est dit *cross-lingue*.

Finalement, en se basant sur la sortie du SRAT, nous distinguons deux grands types de résumés : le résumé extractif et le résumé abstractif. Le *résumé extractif* est généré par la sélection des phrases pertinentes et informatives telles qu'elles apparaissent dans les documents sources. Alors que le *résumé abstractif* (résumé par compréhension) se base sur des techniques qui utilisent une analyse en profondeur de textes pour produire de nouvelles phrases grammaticalement correctes, concises, cohérentes, devant donner un résultat proche d'un résumé humain.

La diversité des sources d'information contenues dans le web a poussé la communauté des chercheurs en résumé automatique de textes à ajouter un autre critère de classification des SRAT : le genre des documents sources. Selon ce facteur, il est également possible d'envisager d'autres types de SRAT à savoir : résumé d'articles de presse, résumé d'un domaine spécialisé (biomédical, droit, etc.), résumé des documents narratifs et de textes littéraires, résumé des pages web, résumé des conversations email, etc. Enfin, il est à noter que ces types de résumés ne sont pas indépendants les uns des autres. Un

résumé textuel qui est rattaché à un type de résumé particulier peut aussi être rattaché à un autre, dans la mesure où il répond à toutes les conditions assurant les fonctions de l'autre type.

2.2 Approches du résumé automatique

Pour générer un résumé automatique de textes, plusieurs méthodes et techniques sont proposées. Principalement, deux grands types d'approches s'opposent : l'approche par extraction et l'approche par abstraction. Plus récemment de nouvelles approches sont apparues, considérées comme semi-extractives mettant en œuvre des techniques de compression, de fusion et de division de phrases. Dans cette section, nous présentons brièvement ces approches.

2.2.1 Approche extractive

Les approches extractives cherchent à repérer et à extraire les segments textuels les plus pertinents pour constituer un résumé. Généralement, le processus de la génération d'un résumé par extraction comporte 4 étapes.

— *Prétraitement* (Analyse et représentation des documents) : les documents sources sont sous une forme non structurée ; cette étape permet de prétraiter ces documents pour les représenter de manière structurée. Le prétraitement implique généralement certaines techniques du TALN, notamment la segmentation de phrases, la tokenisation, la lemmatisation/stemming, la reconnaissance des entités nommées, la résolution de co-référence. Une fois que le prétraitement est terminé, une représentation des documents sources est requise, et elle consiste généralement en la représentation de chaque document par un vecteur, afin de le rendre exploitable par les algorithmes.

— *Pondération de phrases* : cette étape est cruciale pour un SRAT par extraction. En se basant sur la représentation déjà créée dans la première étape et sur des caractéristiques de phrases (Oliveira *et al.*, 2016a), cette phase consiste à assigner à chaque phrase un score indiquant sa pertinence. Plusieurs méthodes sont développées pour cette tâche. Nous citons les plus répandues dans la littérature : méthodes statistiques (Ko & Seo, 2008), méthodes basées sur les graphes (Mihalcea, 2004; Erkan & Radev, 2011; Baralis *et al.*, 2013), méthodes utilisant l'apprentissage automatique (Fattah, 2014; Yang *et al.*, 2014), méthodes utilisant les réseaux de neurones (Nallapati *et al.*, 2017), ainsi que d'autres récentes méthodes.

— *Sélection de phrases* : après avoir calculé les scores des phrases, nous sélectionnons celles ayant un score élevé pour générer un résumé. L'un des problèmes les plus importants de cette étape est d'éviter la redondance, et notamment pour les résumés multi-documents. Pour cela, plusieurs méthodes ont été introduites telles que MMR (Maximum Marginal Relevance) (Carbonell & Goldstein, 1998) et ILP (Integer Linear Programming) (Oliveira *et al.*, 2016b).

— *Génération du résumé* : généralement le système combine les phrases sélectionnées dans l'étape précédente telles qu'elles apparaissent pour générer un résumé.

2.2.2 Approche semi-extractive

La compression, la fusion et la division de phrases sont des axes de recherche relativement récents dans le résumé automatique de textes caractérisant les approches semi-extractives. Ces tâches de traitement des phrases permettent un certain nombre d'améliorations, notamment la réduction de la redondance, et la création de résumés plus proches des résumés abstractifs. *Les approches compressives* consistent à transformer une phrase pertinente en une phrase grammaticalement plus courte qui conserve l'information importante (Knight & Marcu, 2000; Zajic *et al.*, 2008; Torres-Moreno, 2014). *La fusion de phrases* consiste à générer une phrase simple, grammaticalement correcte à partir d'un ensemble de phrases connexes, et qui préserve les informations importantes de cet ensemble. Cette phrase n'est pas obligatoirement contenue dans cet ensemble (Tzouridis *et al.*, 2014; Torres-Moreno, 2014). *La division de phrases* est une nouvelle approche semi-extractive proposée par (Genest & Lapalme, 2011). Cette approche consiste tout d'abord à trouver des Information Items (InIts), qui sont définis comme étant les plus petits éléments d'information cohérents dans une phrase ou dans un texte. Puis, sélectionner ceux qui répondent au besoin d'information de l'utilisateur. Enfin, générer un résumé qui contient les InIts les plus pertinents.

2.2.3 Approche abstractive

Les méthodes de cette approche sont apparues vers la fin des années 1970. Elles s'inspirent principalement du domaine de l'intelligence artificielle et de la psychologie cognitive, et elles cherchent à produire des résumés avec une qualité linguistique comparable à celle des résumés produits par les êtres humains. Généralement, on distingue trois familles de méthodes pour l'approche abstractive (Andhale & Bewoor, 2016) : (i) des méthodes qui traduisent les informations importantes contenues dans les documents sources en des schémas cognitifs tels que les ontologies, les patrons, les graphes, (ii) les méthodes se basant sur la représentation sémantique des documents, et enfin (iii) les méthodes utilisant les réseaux de neurones dans le cadre de l'apprentissage profond (Deep Learning) (Nallapati *et al.*, 2016; Rush *et al.*, 2015).

2.3 Evaluation de systèmes de résumé automatique

L'évaluation de la qualité des résumés automatiques reste toujours une tâche extrêmement subjective et difficile, à laquelle la communauté scientifique a répondu avec des solutions partielles. Les méthodes d'évaluation, telles que la précision et le rappel, qui sont très utilisées dans les systèmes de recherche d'information, ne sont pas vraiment adaptées à cette tâche, vu que les entrées et les sorties des systèmes de résumé automatique sont des textes en langage naturel difficiles à comparer. Les méthodes d'évaluation peuvent être classées en deux types : d'une part les méthodes *intrinsèques* qui évaluent le résumé lui-même en fournissant des mesures automatiques ou semi-automatiques de l'informativité et d'autre part les méthodes *extrinsèques* qui mesurent la qualité du résumé à travers d'autres applications de la fouille de textes.

En ce qui concerne les méthodes *intrinsèques*, citons la méthode *ROUGE* (Lin, 2004) qui est la méthode la plus utilisée, elle est fondée sur la comparaison automatique de n-grammes entre un ou plusieurs résumés de référence et un résumé à évaluer. Il en existe plusieurs variantes, notamment ROUGE-n, ROUGE-SUn et ROUGE-L. La méthode *Pyramide* (Nenkova & Passonneau, 2004) est une autre méthode intrinsèque d'évaluation, mais semi-automatique, ayant pour objectif de surmonter

le problème de sémantique non abordé par ROUGE. Citons aussi la mesure *Responsiveness* qui évalue manuellement le résumé de point de vue du contenu et de la qualité linguistique. Pour les méthodes *extrinsèques*, mesurant la qualité du résumé à travers d'autres applications de la fouille de textes, telles que les systèmes de recherche d'information, la catégorisation de texte et les systèmes Questions-Réponses (Q/R), et qui sont spécifiques à la nature de ces applications.

Pour conclure sur les méthodes d'évaluation, notons que l'évaluation des résumés est une problématique à part entière, à laquelle la campagne TAC a proposé la tâche AESOP (Automatically Evaluating Summaries Of Peers) pour encourager le développement des méthodes automatiques d'évaluation des résumés.

3 Le résumé automatique guidé de textes

Cette section est consacrée à la description du résumé automatique guidé de textes, en se concentrant sur ses spécificités et les contraintes qu'il impose, ainsi que sur l'architecture fonctionnelle générique pouvant lui être associée.

3.1 Définition

La conférence Document Understanding Conference (DUC-2001/2007), et sa remplaçante Text Analysis Conference (TAC) organisées par le NIST (National Institute of Standards and Technology), ont présenté plusieurs types du résumé automatique de textes tels que le résumé orienté, le résumé multi-documents, le résumé de mise à jour. En 2010, la campagne internationale d'évaluation TAC a lancé une nouvelle tâche intéressante et qui représente un grand défi : le résumé automatique guidé de textes. Ce dernier propose un changement significatif vers des résultats orientés sémantiquement tout en favorisant le traitement profond du langage naturel, l'extraction d'information spécifique à un domaine, l'utilisation des ontologies, etc. Et cela dans l'optique d'encourager le passage vers la génération des résumés par abstraction.

La campagne TAC'2010/2011 définit le résumé guidé comme étant un résumé multi-documents de 100 mots obtenu à partir d'un ensemble de 10 articles de presse qui portent sur une thématique précise et appartenant à une catégorie préalablement définie. En l'occurrence, il s'agit d'articles relatifs à des attaques terroristes. Cinq *catégories* ont été sélectionnées et chaque catégorie comporte une liste spécifique *d'aspects* qui sont définis conformément aux thématiques des documents. Selon Jin *et al.* (2011), un aspect est défini comme un thème sémantique représentant un attribut important des entités trouvées dans la collection des documents. Les catégories et leurs aspects ont été développés à partir des modèles des résumés tirés des campagnes DUC/TAC déjà passées. Par exemple, les aspectes de la catégorie ATTACKS sont : WHAT (What Happened), WHEN (date, time, ...), WHERE (location), PERPETRATORS (individuals or groups responsible for the attack), WHY (reasons), WHO AFFECTED (casualities), DAMAGES (caused by the attack) and COUNTERMEASURES (rescues efforts, preventive effort, ...). Le résumé doit couvrir tous ces aspects, comme il peut également contenir d'autres informations pertinentes.

Un système de génération de résumé guidé nécessite également un composant du résumé de mise à jour destiné à produire un résumé sous l'hypothèse que l'utilisateur a déjà lu le résumé des 10 premiers articles, et il n'a besoin que des dernières nouvelles. Alors, le résumé guidé répond à deux

demandes émergentes de traitement de l'information : les exigences en termes d'aspects spécifiques et de temps.

3.2 Architecture fonctionnelle générique

Bien que les détails d'implémentation des systèmes automatiques de résumé guidé soient différents les uns des autres, cependant une architecture fonctionnelle générique illustrée à la figure 1 peut être définie pour les systèmes.

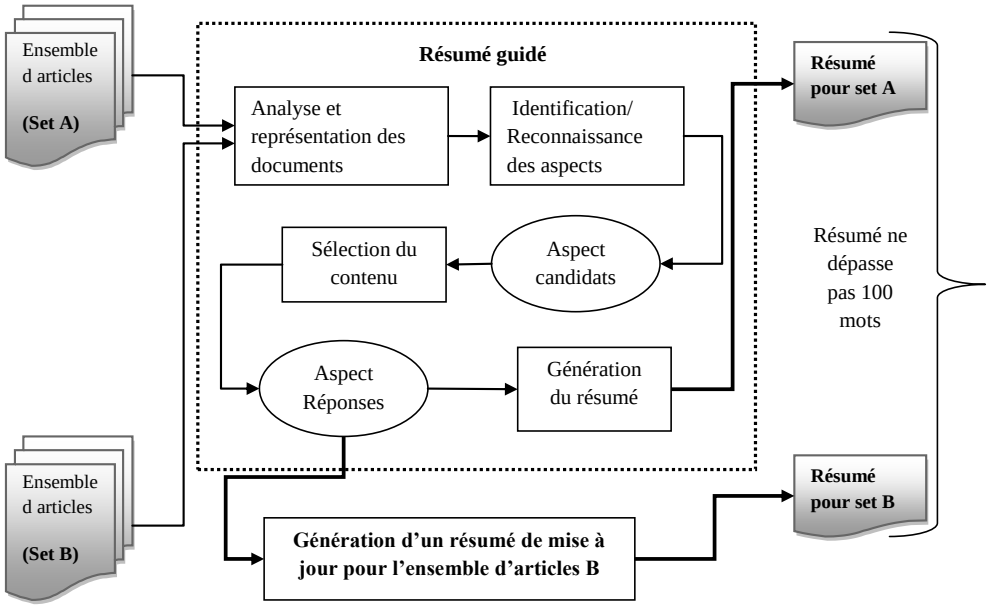


FIGURE 1 – Architecture générale d'un système de résumé guidé

Comme nous l'avons déjà mentionné dans les sections précédentes, un domaine d'application privilégié est le résumé guidé d'articles de presse circulant dans le web. Pour une catégorie d'articles donnée, les articles sont séparés en deux ensembles : *ensemble* A et *ensemble* B. D'abord, le système doit générer un résumé initial pour l'ensemble *d'articles* A, auquel le résumé doit répondre à tous les aspects prédéfinis. Puis, un résumé de mise à jour est généré pour l'ensemble *d'articles* B, en supposant que les documents de *l'ensemble* A ont été lus, et précédent chronologiquement les documents de *l'ensemble* B. Le résumé de mise à jour de *l'ensemble* B est également un résumé guidé, qui ne devrait pas contenir les informations déjà présentées dans le résumé de *l'ensemble* A. Chaque résumé doit être cohérent, concis, bien organisé, contenir des phrases grammaticalement correctes, et ne dépassant pas 100 mots.

Ainsi le résumé guidé peut être considéré comme un résumé multi-documents, de mise à jour, orienté par un ensemble d'aspects. Il regroupe trois types de résumés ce qui affirme que les types de résumé automatique de textes ne sont pas indépendants les uns des autres.

4 Systèmes de résumé automatique guidé

Dans la section 2, nous avons distingué trois grandes approches du résumé automatique en général les approches extractives, semi-extractives et abstractives. Dans cette section, nous présentons tout d'abord de façon succincte différents systèmes de résumé guidé se rapportant à ces trois grandes approches, en précisant les techniques utilisées pour chacun de ces systèmes. Ensuite, nous essayons de les comparer selon différents critères.

4.1 Systèmes basés sur l'approche extractive

Du *et al.* (2010) proposent deux méthodes nommées MRSP (Manifold Ranking with Sink Points) et TMSP (Topic guided Manifold Ranking with Sink Points). La méthode MRSP est dédiée au résumé de mise à jour, et vise à créer des résumés de haute qualité au niveau de la pertinence, l'importance, la nouveauté et la diversité de l'information. La méthode TMSP est une extension du MRSP qui intègre le modèle pLSA (Probablistic Latent Semantic Analysis) (Hofmann, 2013) avec un algorithme d'Espérance Maximum (EM) pour extraire les aspects. Les deux méthodes MRSP et TMSP sont basées sur l'approche Manifold Ranking (Zhou *et al.*, 2003).

Du *et al.* (2011) proposent une nouvelle méthode de classement de phrases DDRank (Decayed DivRank), une extension de DivRank (Mei *et al.*, 2010). Le modèle pLSA (Hofmann, 2013) est utilisé pour attribuer à chaque phrase un score mesurant sa probabilité d'appartenir à certains aspects. DDRank utilise les scores obtenus pour sélectionner les phrases du résumé. DDRank aborde la diversité, la pertinence et l'importance dans le classement de phrases d'une manière unifiée.

Zhang *et al.* (2011) développent deux systèmes de résumé : Polycom1 et Polycom2. Polycom1 est un système de base qui utilise une formule statistique pour attribuer des scores aux phrases sans prendre en considération les scores des aspects. Dans Polycom2 la reconnaissance de l'aspect au niveau de la phrase est considérée comme un problème de classification multilabels de textes auquel le modèle est construit en utilisant un nouveau type de caractéristiques (meta-phrase features), la technique de la décomposition binaire et le SVM (Vapnik, 1998; Joachims, 1999). Le modèle obtenu est ensuite utilisé pour prédire les phrases ayant des aspects et ces informations prédites sont alors utilisées pour calculer les scores des aspects dans les phrases. Polycom2 intègre les scores des aspects avec les scores obtenus par Polycom1 pour calculer les scores finaux des phrases.

Li *et al.* (2011a) proposent deux systèmes pour le résumé guidé : PKTUM1 et PKTUM2. Le premier combine linéairement les scores obtenus par l'algorithme Manifold Ranking avec des scores basés sur d'autres caractéristiques de surface pour pondérer et sélectionner les phrases pertinentes. Le deuxième est basé sur une variante de l'approche ILP (Integer Linear Programming) : T-ILP (Tolerated ILP) qui utilise Wikipedia comme domaine de connaissances pour améliorer la pondération des concepts. L'étape de sélection de phrases est précédée par un prétraitement et suivie par un post-traitement.

Barrera *et al.* (2011) proposent une méthode pour la génération des résumés guidés qui utilise un système Questions/Réponses SemQuest pour répondre aux aspects prédéfinis pour chaque catégorie.

Steinberger *et al.* (2010) proposent une nouvelle méthode pour le résumé guidé qui combine un système d'extraction d'information NEXUS (Tanev *et al.*, 2008) et la méthode LSA (Latent Semantic Analysis) (Gong & Liu, 2001; Steinberger & Jezek, 2004) pour capturer les informations pertinentes spécifiques aux aspects.

Varma *et al.* (2010) proposent une approche pour le résumé guidé qui se base sur un système d'extraction d'information et qui comporte principalement quatre étapes : la construction du domaine de connaissances, l'annotation de phrases pour identifier les informations spécifiques aux aspects, l'extraction des concepts pour calculer l'importance des phrases, et enfin la génération du résumé.

Conroy *et al.* (2011) développent le système CLASSY pour un résumé guidé multilingue qui combine des techniques statistiques avec le modèle naïf bayésien pour pondérer et sélectionner les informations pertinentes.

Zhang *et al.* (2012) améliorent le système qu'ils ont proposé en TAC'2011 (Zhang *et al.*, 2011), en ajoutant un modèle HMM (Hidden Markov Model) pour maximiser la cohérence des aspects trouvés.

Ng *et al.* (2012) proposent le système SWING (Guided Summarizer from WING "The Web Information Retrieval/Natural Language Processing") pour le résumé guidé qui se base sur les principes des systèmes de recherche d'information pour faire l'extraction des phrases pertinentes. L'idée principale est l'utilisation de la connaissance de catégorie (Category Knowledge), pour calculer l'importance spécifique des phrases par rapport à la catégorie (Category specific importance CSI). Le système Swing est classé parmi les meilleurs systèmes dans la campagne TAC'2011, et il pourra être appliqué à n'importe quelle catégorie, car les scores des caractéristiques utilisées se calculent à trois niveaux : le corpus, la catégorie et la thématique.

4.2 Systèmes basés sur l'approche semi-extractive

Li *et al.* (2011b) proposent une nouvelle approche par compression de phrases qui se divise en quatre étapes à savoir le clustering, le classement, la compression et la sélection de phrases. La première étape utilise un modèle LDA (Latent Dirichlet Allocation) pour identifier automatiquement les différents aspects et pour regrouper les phrases en ces aspects. La deuxième étape utilise une extension de l'algorithme LexRank (Erkan & Radev, 2004) pour classer les phrases dans chaque cluster. La troisième étape utilise un nouvel algorithme de compression de phrases pour améliorer la qualité linguistique des résumés. Enfin, la dernière étape utilise un Framework de programmation linéaire entière (ILP) pour sélectionner les phrases pertinentes et qui répondent aux aspects.

4.3 Systèmes basés sur l'approche abstractive

Peu de systèmes de résumé guidé existe, citons notamment le système ABSUM (Genest & Lapalme, 2012). Ce système implémente l'approche K-BABS (Knowledge-Based ABstractive Summarization) pour la génération automatique des résumés guidés par abstraction. L'architecture se constitue principalement de trois modules : le premier réalise une analyse fournissant une représentation intermédiaire des documents sources riche en information syntaxique et sémantique. Un deuxième module élabore un plan, appelé Task Blueprint qui définit manuellement les règles d'extraction d'information pour trouver les aspects candidats à partir de la représentation. Enfin,un troisième module fait une sélection de contenu sur les aspects candidats pour sélectionner les aspects réponses et générer ensuite le résumé par l'usage du logiciel SimpleNLG realizer (Gatt & Reiter, 2009).

4.4 Analyse comparative des systèmes de résumé automatique guidé

Dans cette sous-section, les systèmes brièvement présentés sont comparés selon divers critères, conduisant aux deux tableaux. Le premier tableau compare ces systèmes selon l'approche préconisée et les techniques mises en œuvre. Le second tableau compare ces systèmes selon les datasets qui ont été traités dans les publications les présentant, les mesures d'évaluation utilisées, les campagnes d'évaluation auxquelles ils ont participé, et enfin les résultats obtenus par ces systèmes dans ces campagnes.

TABLE 1 – comparaison des systèmes selon l'approche adoptée et les méthodes utilisées

Systèmes	Approche adoptée	Méthodes utilisées
Du *et al.* (2010)	extractive statistique	Algorithme Manifold Ranking - TMSP - MRSP - pLSA - l'algorithme Espérance-Maximisation
Steinberger *et al.* (2010)	extactive basée sur un système d'extraction d'événements	Le système NEXUS d'extraction - d'événements - LSA
Du *et al.* (2011)	extractive statistique	Algorithme DivRank - algorithme Decayed DivRank - pLSA
PolyCom (Zhang *et al.*, 2011)	extractive utilisant l'apprentissage automatique	SVM- Décomposition linéaire - (ISF) Inverted sentence frequency
PKTUM1/ PKTUM2 (Li *et al.*, 2011a)	extractive statistique	Algorithme Manifold Ranking ILP (Integer Linear Programming) - T-ILP (Tolerated-ILP)
Li *et al.* (2011b)	semi-extractive par compression de phrases utilisant l'apprentissage non-supervisé	Clustering - LDA (Latent Dirichlet Location) - l'algorithme LexRank - ILP (Integer Linear Programming)
CLASSY 2011 (Conroy *et al.*, 2011)	extractive hybride utilisant des techniques statistiques et l'apprentissage automatique	Naïf bayésien - caractéristiques linguistiques et statistiques
SWING (Ng *et al.*, 2012)	extractive hybride utilisant des techniques statistiques et l'apprentissage automatique	SVR - CRS (category relevance score) - CKLD (Category KL-divergence score) - INDF (Interpolated N-gram document frequency) - MMR
ABSUM (Genest & Lapalme, 2012)	abstractive (Template-based approach)	Analyse syntaxique et sémantique - Extraction de l'Information - NLG (Natural Language Generation)

D'après l'étude que nous avons menée sur les systèmes du résumé guidé, il est clair que *l'approche extractive* est celle qui a été retenue par la majorité des meilleurs systèmes. Rappelons que le principe de base de cette approche consiste à extraire des phrases pertinentes correspondantes aux aspects définis pour chaque catégorie définie pour le résumé guidé. Généralement, les premiers systèmes du résumé guidé exploitaient des méthodes statistiques, des méthodes d'apprentissage automatique, et des méthodes fondées sur la programmation linéaire avec une analyse de surface. Les résultats obtenus sont généralement intéressants, par exemple le système SWING (Ng *et al.*, 2012) fondé sur des méthodes statistiques est classé parmi les meilleurs systèmes dans TAC'2011. Le système CLASSY (Conroy *et al.*, 2011) basé à la fois sur des méthodes statistiques et sur un classificateur naïf bayésien est classé le premier en termes de *Overall Responsiviness*, et il atteint des scores intéressants par les autres méthodes d'évaluation ROUGE et Pyramide. En adoptant la même approche, d'autres systèmes améliorés ont été proposés, en introduisant des ressources sémantiques externes telles que Wordnet et Wikipedia, citons notamment le système PKTUM2 (Li *et al.*, 2011a). Basé sur une variante de la méthode ILP (Integer Linear Programming), celui-ci exploite Wikipidea comme

ressource sémantique, et il est classé premier par la méthode d'évaluation Pyramide et deuxième en termes de *Overall Responsiviness*.

D'autres auteurs (Li *et al.*, 2011b) ont opté pour *l'approche semi-extractive* en s'appuyant sur des techniques de compression de phrases, l'idée sous-jacente est de résoudre une des majeures problématiques du résumé par extraction, en éliminant les informations superflues et non essentielles contenues dans les phrases extraites. De plus elle permet d'établir un pont vers le résumé par abstraction. Le système proposé par Li *et al.* (2011b) a assuré des résultats intéressants.

L'approche abstractive constitue l'approche la plus récente, et la plus difficile à mettre en œuvre. Dans ce contexte, le système ABSUM proposé par (Genest & Lapalme, 2012) a atteint des résultats satisfaisants en termes de la qualité linguistique et du score Content Density. Cependant, les résumés générés par ABSUM se composent d'une moyenne de 21 mots contrairement aux 100 mots générés par les autres systèmes. Pour pallier cette limitation et améliorer le score du Overall Responsiveness du résumé, Genest & Lapalme (2012) ont introduit une approche hybride qui combine le système ABSUM et le système CLASSY. Les résultats obtenus montrent une amélioration significative en termes du Overall Responsiveness.

D'autres travaux (Barrera *et al.*, 2011) adoptent un système Questions-Réponses pour répondre aux aspects prédéfinis pour chaque catégorie. Une autre tendance consiste à aborder la problématique du résumé guidé comme plutôt un problème d'extraction d'information. Le système proposé par Varma *et al.* (2010), basé sur un système d'extraction d'information, a atteint le premier rang selon les méthodes ROUGE-2, ROUGE-SU4 et Pyramide, et il est classé le deuxième en termes du *Overall Responsiveness*.

5 Conclusion et perspectives

De l'analyse comparative précédente, on constate que les approches extractives sont actuellement dominantes dans le développement de systèmes automatiques de résumé guidé. Le principal avantage des méthodes extractives est qu'elles sont relativement simples à mettre en œuvre et qu'elles ne nécessitent pas une analyse en profondeur des textes, analyse assez complexe. L'inconvénient principal de ces méthodes est que les résumés produits manquent souvent de cohérence et de cohésion. En ce qui concerne les approches semi-abstractives, plus récentes, elles essaient de compenser les faiblesses des approches extractives sans pour autant les remettre en cause. Enfin les approches abstractives, bien que prometteuses, sont très difficiles à mettre en œuvre dans des systèmes automatiques de résumé. D'une façon générale, pour améliorer la qualité des résumés guidés obtenus par ces systèmes, il nous semble nécessaire de prendre en compte plus de sémantique, sémantique liée soit au domaine d'application du résumé guidé recherché, caractérisé par les catégories et leurs aspects, soit liée à la langue naturelle dans laquelle sont écrits les textes en entrée (source) et en sortie (si multilingue). Pour cela nous avons identifié deux grandes voies de recherche :

1. *Pour les approches extractives et semi-abstractives* : une amélioration majeure consisterait à adopter des représentations vectorielles enrichies des documents sources plus sémantiques, en s'appuyant notamment sur la désambiguïsation du sens (WSD), le calcul de similarité sémantique entre les mots. Cette approche est déjà utilisée dans le système développé par (Plaza *et al.*, 2010) pour résumer des documents biomédicaux, l'utilisation de la WSD permet d'améliorer les résultats obtenus en termes de performance. La construction des représentations lexicales distribuées pourrait également améliorer les systèmes du résumé. Ce type de représentations

peut se baser sur des techniques de réseaux de neurones dans le cadre de l'apprentissage profond. Ces techniques ont montré des résultats très intéressants en traitement automatique de la langue naturelle (Luong *et al.*, 2013; Zheng *et al.*, 2013), mais dans le résumé automatique très peu de travaux les utilisent jusqu'à présent, citons, cependant les travaux développés dans (Denil *et al.*, 2014).

2. *Pour l'approche abstractive* : ces approches utilisent déjà des représentations des documents sources plus sémantiques, il s'agirait principalement d'automatiser certaines tâches de traitement, actuellement réalisées de façon manuelle, ceci par la mise en œuvre de techniques récentes d'apprentissage machine. L'intérêt de l'apprentissage symbolique pour la réalisation de ces tâches est qu'il se situe au même niveau sémantique que celui pouvant être associé d'une part aux catégories et leurs aspects pouvant être liés à des connaissances externes comme des ontologies, et d'autre part à des connaissances linguistiques spécifiques. Ainsi, considérant le système (Genest & Lapalme, 2012), sa tâche de *blueprint* réalisant l'extraction des informations relatives aux aspects est actuellement réalisée de façon manuelle. Elle ne couvre pas toutes les catégories et elle nécessite assez de temps et d'efforts humains. Son automatisation, conduisant à l'automatisation de ce processus d'extraction d'information pouvant exploiter des ontologies, améliorerait de façon considérable ce système.

La production automatique de résumés guidés par abstraction reste un domaine jusqu'à présent peu exploré. Aux grand défis liés à la cohérence, la cohésion et la lisibilité des résumés générés, l'approche abstractive semble cependant la plus prometteuse. Très peu de systèmes ont été crées selon cette approche, citons FRUMP (DeJong, 1982), RIPTIDES (White *et al.*, 2001), et ABSUM (Genest & Lapalme, 2012). Tous ces systèmes combinent l'extraction d'information et les techniques de génération automatique de textes, mais reposent sur des tâches manuelles, notamment au niveau de l'extraction d'information.

Dans ce contexte, dans l'objectif de developper des systèmes automatiques de résumé guidé performants, notre recherche s'oriente plutôt dans la seconde voie, et aurait comme objectif la proposition d'une méthode abstractive pour le résumé guidé, composée de quatre étapes : (1) Analyse et la représentation des documents, (2) Extraction d'information, (3) Sélection de contenu, et (4) Génération du résumé. Cette méthode exploiterait une ontologie de domaine comme ressource sémantique externe pour guider le processus d'extraction d'information automatique, conformément à l'objectif du résumé guidé, l'utilisation des ontologies rend le contenu de résumé centré sur les besoins de l'utilisateur (Mohan *et al.*, 2016). L'étape d'*analyse et de la représentation des documents* se baserait sur des techniques linguistiques profondes qui exploiteraient la structure discursive de texte, notamment la théorie de la structure rhétorique (Mann & Thompson, 1988). L'étape d'*extraction d'information* reposerait sur l'utilisation d'une ontologie de domaine et sur une technique d'apprentissage automatique pour capturer les informations spécifiques aux aspects, comme le fait le système OntoILPER (Espinasse *et al.*, 2016) en utilisant la programmation logique inductive. L'étape de *sélection de contenu* exploiterait des ressources sémantiques externes (ontologies). Enfin, l'étape de *génération du résumé* serait assurée par le logiciel SimpleNLG Realizer (Gatt & Reiter, 2009), en prenant en considération l'importance des aspects et la relation entre eux. Une combinaison des approches abstractives et (semi)-extractives pourrait aussi être localement considérée.

TABLE 2 – Comparaison des principaux systèmes de résumé guidé

Systèmes	Datasets utilisés	Mesures d'évaluation	Campagne	Résultats
(Du *et al.*, 2010)	Apprentissage : TAC'2008/2009 test : TAC'2010	ROUGE-2 (R-2) ROUGE-SU4 (R-SU4) Pyramid Basic element (BE)	Les SRAT guidés participants à la compétition organisée par la campagne TAC'2010	Pour TMSP : Pyramid-A= 0,351 Rank (18), BE-A= 0.04529 Rank(21), R-2-A= 0.07623 Rank (21), R-SU4-A = 0.11042 Rank(21) Pour MRSP : Pyramid-B= 0,276 Rank(3), BE-B=0.04350 Rank(3), R-2-B=0.07251 Rank(4), R-SU4 - B = 0.11039 Rank(5)
(Du *et al.*, 2011)	TAC'2011	ROUGE-2 (R-2) ROUGE-SU4 (R-SU4) Pyramid Basic element (BE)	Les SRAT guidés participants à la compétition organisée par la campagne TAC'2011	Pour set A : Pyramid= 0.435 Rank (14) BE= 0.07099 Rank(13), R-A= 0.11324 Rank(11) R-SU4-A = 0.14901 Rank (9), Pour set B : Pyramid = 0.335 Rank(3), BE-B=0.05717 Rank(3), R-2 = 0.07992 Rank(14), R-SU4 = 0.12062 Rank(5)
PolyCom (aspect-integrated system) (Zhang *et al.*, 2011)	Apprentissage : TAC'2010 et un corpus crée manuellement à partir de DUC/TAC déjà passés Test : TAC'2011	ROUGE-2 (R-2) ROUGE-SU4 (R-SU4) Pyramid Basic element (BE) Linguistic Quality (LQ)	Les SRAT guidés participants à la compétition organisée par la campagne TAC'2011	Pour set A : R-2=0.12306 Rank (4), R-SU=0.15975 Rank(3), BE=0.07938 Rank(4), Pyramid=0,437 Rank(8), LQ=2,932 Rank(26) Pour set B : R-2=0.08643 Rank (4), R-SU=0.12803 Rank(2), BE=0.05437 Rank(9), Pyramid=0.3 Rank(17), LQ=2.795 Rank(25)
PKTUM1/ PKTUM2 (Li *et al.*, 2011a)	TAC'2011	ROUGE-2 (R-2) ROUGE-SU4 (R-SU4) Pyramid Basic element (BE) Linguistic Quality (LQ) Overall Responsiveness (OR)	Les SRAT guidés participants à la compétition organisée par la campagne TAC'2011	Pour PKTUM1 set A : R-2=0.102, R-SU=0.15975 Pyramid=0,418, LQ=3,136, OR= 2,977 (Rank13) Pour set B : R-2=, 0.0709, R-SU= 0.114, Pyramid=0.264, LQ= 3.023, OR= 2.432 (Rank15) Pour PKTUM2 set A : R-2= 0.115, R-SU= 0.150, Pyramid= 0.477 (Rank 1), LQ= 3.432, OR= 3.136 (Rank2) Pour set B : R-2= 0.0816, R-SU= 0.119, Pyramid= 0.313 LQ= 3.273, OR= 2.477 (Rank11)
(Li *et al.*, 2011b)	TAC'2010	ROUGE-1 ROUGE-2, ROUGE-L (R-L) ROUGE-SU4 ROUGE-W-1.2 (R-W-1.2)	K-means entity-aspect, greedy1, greedy2, KL-Div HIERSUM (Haghighi and Vanderwende, 2009),	R-1 = 0.32641, R-2= 0.06508,R-SU4= 0.10146, R-W-1.2 = 0.09998, R-L = 0.28610
CLASSY 2011 (Conroy *et al.*, 2011)	TAC'2011	Overall Responsiveness (OR) Linguistic Quality (LQ) Pyramid Content Density (CD) size	Les SRAT guidés participants à la compétition organisée par la campagne TAC'2011	Pyramid= 0.520, LQ= 3.39, OR= 3.20 Rank(1), size = 98.0%, CD= 0.0053
SWING (Ng *et al.*, 2012)	TAC'2011	ROUGE-2 ROUGE-SU4	Generic+ CRS, Generic +CKLD, CLASSY, PolyCom	R-2 = 0.13796, R-SU4 = 0.16808
ABSUM (Genest & Lapalme, 2012)	TAC'2011 Catégories : Attacks, Accidents and Natural disasters	Overall Responsiveness (OR) Linguistic Quality (LQ) Pyramid Content Density (CD) size	ABSUM, CLASSY 2011, ABSUM/CLASSY Hybrid, Extractive baseline, Abstractive baseline (Genest et Laplme, 2011), Human-written models	Pour ABSUM : Pyramid= 0.277 Rank (5), LQ= 3.67 (Rank 3), OR= 2.07 Rank(5), size = 22,6 % Rank(6), CD= 0.0119 Rank (1) Pour ABSUM/CLASSY hybrid : Pyramid= 0.600 Rank (1), LQ= 3.28 (Rank 4), OR= 3.31 Rank(2), size = 97.6% Rank(3), CD= 0.0061 Rank (2)

Références

ANDHALE N. & BEWOOR L. (2016). An overview of text summarization techniques. In *Computing Communication Control and automation (ICCUBEA), 2016 International Conference on*, p. 1–7 : IEEE.

BARALIS E., CAGLIERO L., MAHOTO N. A. & FIORI A. (2013). Graphsum : Discovering correlations among multiple terms for graph-based summarization. *Inf. Sci.*, **249**, 96–109.

BARRERA A., VERMA R. M. & VINCENT R. (2011). Semquest : University of houston's semantics-based question answering system. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*.

CARBONELL J. & GOLDSTEIN J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, p. 335–336 : ACM.

CONROY J. M., SCHLESINGER J. D., KUBINA J., RANKEL P. A. & O'LEARY D. P. (2011). CLASSY 2011 at TAC : guided and multi-lingual summaries and evaluation metrics. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*.

CREMMINS E. (1993). Valuable and meaningful text summarization in thoughts, words, and deeds. In *Workshop on Summarizing Text for Intelligent Communication*.

CREMMINS E. (1996). *The Art of Abstracting*. Information Resources Press.

DEJONG G. (1982). An overview of the FRUMP system. In W. LEHNERT & M. RINGLE, Eds., *Strategies for Natural Language Processing*, p. 149–176. Lawrence Erlbaum.

DENIL M., DEMIRAJ A., KALCHBRENNER N., BLUNSOM P. & DE FREITAS N. (2014). Modelling, visualising and summarising documents with a single convolutional neural network. *CoRR*, **abs/1406.3830**.

DU P., YUAN J., LIN X., ZHANG J., GUO J. & CHENG X. (2011). Decayed divrank for guided summarization. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011*.

DU P., ZHANG J., GUO J. & CHENG X. (2010). TMSP : topic guided manifold ranking with sink points for guided summarization. In *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*.

ERKAN G. & RADEV D. R. (2004). Lexrank : Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, **22**, 457–479.

ERKAN G. & RADEV D. R. (2011). Lexrank : Graph-based lexical centrality as salience in text summarization. *CoRR*, **abs/1109.2128**.

ESPINASSE B., LIMA R. & FREITAS F. (2016). Extraction automatique d'entités et de relations par ontologies et programmation logique inductive. *Revue d'Intelligence Artificielle*, **30**(6), 637–674.

FATTAH M. A. (2014). A hybrid machine learning model for multi-document summarization. *Appl. Intell.*, **40**(4), 592–600.

GATT A. & REITER E. (2009). Simplenlg : A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, p. 90–93 : Association for Computational Linguistics.

GENEST P. & LAPALME G. (2012). Fully abstractive approach to guided summarization. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2 : Short Papers*, p. 354–358.

GENEST P., LAPALME G. & MONOD M. Y. (2009). HEXTAC : the creation of a manual extractive run. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*.

GENEST P.-E. & LAPALME G. (2011). Framework for abstractive summarization using text-to-text generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, p. 64–73 : Association for Computational Linguistics.

GONG Y. & LIU X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *SIGIR 2001 : Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, p. 19–25.

HOFMANN T. (2013). Probabilistic latent semantic analysis. *CoRR*, **abs/1301.6705**.

JIN F., HUANG M. & ZHU X. (2011). Guided structure-aware review summarization. *J. Comput. Sci. Technol.*, **26**(4), 676–684.

JOACHIMS T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, p. 200–209.

KNIGHT K. & MARCU D. (2000). Statistics-based summarization - step one : Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, p. 703–710.

KO Y. & SEO J. (2008). An effective sentence-extraction technique using contextual information and statistical approaches for text summarization. *Pattern Recognition Letters*, **29**(9), 1366–1371.

LI H., HU Y., WAN X., XIAO J. & LI Z. (2011a). PKUTM participation at TAC 2011 summarization track. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*.

LI P., WANG Y., GAO W. & JIANG J. (2011b). Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, p. 1137–1146.

LI S., WANG W. & ZHANG Y. (2009). TAC 2009 update summarization of ICL. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In S. S. MARIE-FRANCINE MOENS, Ed., *Text Summarization Branches Out : Proceedings of the ACL-04 Workshop*, p. 74–81, Barcelona, Spain : Association for Computational Linguistics.

LLORET E. & PALOMAR M. (2012). Text summarisation in progress : a literature review. *Artif. Intell. Rev.*, **37**(1), 1–41.

LUHN H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, **2**(2), 159–165.

LUONG T., SOCHER R. & MANNING C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, p. 104–113.

MANN W. C. & THOMPSON S. A. (1988). Rhetorical structure theory : Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, **8**(3), 243–281.

MEI Q., GUO J. & RADEV D. (2010). Divrank : the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 1009–1018 : Acm.

MIHALCEA R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21-26, 2004 - Poster and Demonstration*.

MOHAN M. J., SUNITHA C., GANESH A. & JAYA A. (2016). A study on ontology based abstractive summarization. *Procedia Computer Science*, **87**, 32–37.

NALLAPATI R., ZHAI F. & ZHOU B. (2017). Summarunner : A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, p. 3075–3081.

NALLAPATI R., ZHOU B., DOS SANTOS C. N., GÜLÇEHRE Ç. & XIANG B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, p. 280–290.

NENKOVA A. & MCKEOWN K. (2012). A survey of text summarization techniques. In *Mining Text Data*, p. 43–76.

NENKOVA A. & PASSONNEAU R. J. (2004). Evaluating content selection in summarization : The pyramid method. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, p. 145–152.

NG J., BYSANI P., LIN Z., KAN M. & TAN C. L. (2012). Exploiting category-specific information for multi-document summarization. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference : Technical Papers, 8-15 December 2012, Mumbai, India*, p. 2093–2108.

OLIVEIRA H., FERREIRA R., LIMA R., LINS R. D., FREITAS F., RISS M. & SIMSKE S. J. (2016a). Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Syst. Appl.*, **65**, 68–86.

OLIVEIRA H., LIMA R., LINS R. D., FREITAS F., RISS M. & SIMSKE S. J. (2016b). Assessing concept weighting in integer linear programming based single-document summarization. In *Proceedings of the 2016 ACM Symposium on Document Engineering, DocEng 2016, Vienna, Austria, September 13 - 16, 2016*, p. 205–208.

PLAZA L., STEVENSON M. & DÍAZ A. (2010). Improving summarization of biomedical documents using word sense disambiguation. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, p. 55–63 : Association for Computational Linguistics.

RUSH A. M., CHOPRA S. & WESTON J. (2015). A neural attention model for abstractive sentence summarization. *CoRR*, **abs/1509.00685**.

SPARCK J. K. (1998). Automatic summarising : factors and directions. *CoRR*, **cmp-lg/9805011**.

STEINBERGER J. & JEZEK K. (2004). Text summarization and singular value decomposition. In *Advances in Information Systems, Third International Conference, ADVIS 2004, Izmir, Turkey, October 20-22, 2004, Proceedings*, p. 245–254.

STEINBERGER J., TANEV H., KABADJOV M. A. & STEINBERGER R. (2010). Jrc's participation in the guided summarization task at TAC 2010. In *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*.

TANEV H., PISKORSKI J. & ATKINSON M. (2008). Real-time news event extraction for global crisis monitoring. In *Natural Language and Information Systems, 13th International Conference on Applications of Natural Language to Information Systems, NLDB 2008, London, UK, Proceedings*, p. 207–218.

TORRES-MORENO J.-M. (2014). *Automatic text summarization*. John Wiley & Sons.

TZOURIDIS E., NASIR J. & BREFELD U. (2014). Learning to summarise related sentences. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics : Technical Papers*, p. 1636–1647.

VAPNIK V. (1998). *Statistical learning theory*. Wiley.

VARMA V., BYSANI P., B K. R., REDDY V. B., KOVELAMUDI S., VADDEPALLY S. R., NANDURI R., KUMAR N. K., GSK S. & PINGALI P. (2010). IIIT hyderabad in guided summarization and knowledge base population. In *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*.

WHITE M., KORELSKY T., CARDIE C., NG V., PIERCE D. & WAGSTAFF K. (2001). Multidocument summarization via information extraction. In *Proceedings of the first international conference on Human language technology research*, p. 1–7 : Association for Computational Linguistics.

YANG L., CAI X., ZHANG Y. & SHI P. (2014). Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Inf. Sci.*, **260**, 37–50.

ZAJIC D. M., DORR B. J. & LIN J. (2008). Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing & Management*, **44**(4), 1600–1610.

ZHANG R., LI W. & GAO D. (2012). Generating coherent summaries with textual aspects. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

ZHANG R., YOU O. & LI W. (2011). Guided summarization with aspect recognition. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*.

ZHENG X., CHEN H. & XU T. (2013). Deep learning for chinese word segmentation and pos tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 647–657.

ZHOU D., WESTON J., GRETTON A., BOUSQUET O. & SCHÖLKOPF B. (2003). Ranking on data manifolds. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS]*, p. 169–176.

Identification de descripteurs pour la caractérisation de registres

Jade Mekki[1,2] Delphine Battistelli[2] Gwénolé Lecorvé[1] Nicolas Béchet[3]
(1) Univ Rennes, CNRS, IRISA, 6, rue de Kerampont, 22305 Lannion Cedex, France
(2) Universite Paris-Ouest-Nanterre, MoDyCo, 200, avenue de la République 92001 Nanterre Cedex, France
(3) Universite de Bretagne Sud, IRISA, Campus de Tohannic, rue Yves Mainguy, 56017 Vannes Cedex, France
prenom.nom@irisa.fr, delphine.battistelli@u-paris10.fr

RÉSUMÉ
L'article présente une étude des descripteurs linguistiques pour la caractérisation d'un texte selon son registre de langue (familier, courant, soutenu). Cette étude a pour but de poser un premier jalon pour des tâches futures sur le sujet (classification, extraction de motifs discriminants). À partir d'un état de l'art mené sur la notion de registre dans la littérature linguistique et sociolinguistique, nous avons identifié une liste de 72 descripteurs pertinents. Dans cet article, nous présentons les 30 premiers que nous avons pu valider sur un corpus de textes français de registres distincts.

ABSTRACT
Feature identification for register characterization.

The paper presents a study of linguistic features for the characterization of a text according to its language register (formal, neutral, informal). This study aims at laying a first milestones for future work on this subject (e.g., classification, discriminating patterns extraction, etc.). From a state of the art conducted on the notion of register in linguistics and sociolinguistics, we have identified a list of 72 relevant descriptors. In this paper, we present the first 30 ones that we could validate on a corpus of French texts from distinct registers.

MOTS-CLÉS : registres de langue, descripteur linguistique, validation.

KEYWORDS: language register, linguistic feature, validation.

1 Introduction

Cet article présente les premiers travaux liés à un projet plus vaste dont la finalité est d'identifier automatiquement le registre d'un texte, puis de générer des paraphrases qui le transposent vers un registre différent. La caractérisation automatique d'un registre n'a, à notre connaissance, pas fait l'objet de travaux en TAL. Nous situant pour notre part dans cette perspective, une première brique de notre travail consiste à pouvoir disposer d'une première liste de descripteurs suffisamment exhaustive et validée en corpus. Nous exposons donc un travail de recherche exploratoire qui tend à identifier des descripteurs linguistiques pour différents registres (familier, courant et soutenu). C'est cette première brique que nous décrivons ici. L'analyse s'appuie conjointement sur un travail d'expertise et des résultats statistiques. Cette phase préparatoire nous a permis d'associer certains motifs à un registre particulier : nous tendons simplement à exposer les descripteurs étudiés ainsi que leurs comportements en contexte.

Nous commençons ici par revenir sur la notion même de registre en linguistique (section 2), avant de

présenter notre méthodologie d'analyse et de validation ainsi que notre corpus (section 3). Quelques descripteurs sont détaillés (section 4) avant de donner les résultats en (section 5), accompagnés d'une discussion.

2 La notion de registre de langue

Chaque production linguistique est évaluée par l'interlocuteur. Il la caractérise en la situant dans une classe, un registre. Ce dernier permet de qualifier une certaine actualisation de la langue. Ainsi, la notion de registre de langue se situe à la jonction de la linguistique et de la sociolinguistique. Elle s'entend globalement comme renvoyant la variété linguistique associée à une situation de communication particulière, indépendamment de paramètres liés au locuteur/scripteur comme, par exemple, son origine sociale ou son état émotionnel, et se caractérise par des patrons linguistiques spécifiques (Ferguson, 1982; Ledegen & Léglise, 2013). Néanmoins, les travaux du domaine mettent en exergue une difficulté définitoire. Ainsi, les dénominations « niveau », « style » ou encore « genre » co-existent avec celle de « registre » et font l'objet de débats (Bell, 1984; Biber & Finegan, 1994; Gadet, 1996). Ainsi, le partitionnement de l'espace linguistique en registres va varier selon l'angle d'étude. Par exemple, Ilmola (2012) mettra l'accent sur les registres familier, populaire et vulgaire dans des journaux satiriques, là où Borzeix & Fraenkel (2005) s'intéresseront à catégoriser différentes situations de communication au travail pour mettre notamment en exergue la manière dont les mots permettent de mettre en place une forme de contestation des normes établies. De fait, quand on aborde la notion de « registre », la question de (l'écart à) la norme apparaît comme centrale.

En effet, la perception d'un registre passe par l'évaluation de la façon de parler d'un locuteur. Bien que le sujet de notre étude ne soit pas de discuter des ressorts qui associent telle production avec telle connotation, il existe de fait une évaluation axiologique de chaque production linguistique : une forme de jugement de valeur assortie d'une hiérarchisation inhérente à toute évaluation de production linguistique.

Cette notion de hiérarchie nous amène à interroger celle de la « norme » : comment pouvons nous la définir ? Le registre « courant » semble représenter ce que nous appelons la norme. Effectivement, instinctivement nous pourrions penser qu'elle est constituée de toutes les productions linguistiques qui suivent correctement les règles grammaticales françaises. Cependant, une seconde interprétation de la norme serait qu'elle ne produise pas l' évaluation mais qu'au contraire, elle la renforce grâce à son mécanisme de rationalisation. Autrement dit, elle donnerait des justifications a posteriori de ce que nous avons perçu comme étant une variation. La seconde proposition permet de dépasser la valeurs normative a priori d'un bon usage de la langue. Cependant, cette définition est difficilement formalisable en traitement automatique des langues : c'est pourquoi nous adoptons la première proposition.

Par ailleurs, il est souvent admis que le partitionnement en registres distincts est davantage une commodité théorique qu'une réalité de terrain, les différentes pratiques de la langue s'exprimant généralement selon un continuum et non des oppositions tranchées (Blanche-Benveniste, 1997). Notre travail ne visant pas – en tout cas de manière directe – à une contribution sur ces questions, nous adoptons volontairement une approche consensuelle en employant le terme « registre », issu de la tradition britannique (Ure, 1982; Sanders, 1993), et en distinguant trois grands registres communément admis : familier, courant et soutenu. Si nous constatons et admettons qu'un continuum existe, nous devons en effet envisager la notion de registre avec des valeurs discrètes afin d'appréhender cette

notion d'un point de vue automatique.

Un examen approfondi de la littérature linguistique et sociolinguistique nous a permis de répertorier un nombre relativement important de caractéristiques linguistiques, de nature différente (lexicale, syntaxique, etc.), classiquement retenues et sur lesquelles nous reviendrons plus en détail dans la section 5. Les exemples (1), (2), (3), (4) donnent un aperçu de la complexité plus ou moins grande qu'il peut y avoir à envisager pour classifier automatiquement des phrases ou des textes dans un registre.

(1) *Moi, les enquêtes de terrain, bof.* tiré de (Frei, 1929)

(2) *Aussi dément que cela paraisse, il prend au sérieux le droit à l'alimentation, sur une terre où 1,2 milliard [sic] de couillons loin de chez nous souffrent d'une faim chronique.* tiré de (Ilmola, 2012).

(3) *Vous détenez un petit compte bancaire aux Bahamas ou à Jersey ? Comment puiser discrètement, et à distance, dans ce magot sans être tracassé ?* tiré de (Ilmola, 2012).

(4) *Le conseil général la saque non pas parce qu'elle gagne un peu trop, mais au contraire pour ne pas avoir atteint un « revenu minimum » d'au moins 701 euros par mois avec son activité d'autoentrepreneur !* tiré de (Ilmola, 2012).

Dans ces quatre exemples seul un terme par phrase est du registre familier voire vulgaire : « bof », « magot », « saque », « couillons ». Il est intéressant de noter que le registre neutre oscille avec le soutenu car nous trouvons la forme « cela » dans le troisième exemple, ainsi que des planificateurs de discours dans le deuxième : « non pas parce qu'... », « mais au contraire pour... ». En outre, il est important de mettre en regard le but de ces citations et leur lexique : ce sont des extraits tirés de journaux satiriques qui tendent à faire rire leurs lecteurs. Or l'effet comique naît d'une rupture de l'horizon d'attente du lecteur (Bergson, 2013). Dès lors, le décalage entre le genre neutre voire soutenu de la phrase avec un élément lexical du registre familier voire vulgaire tend à opérer cette rupture afin de créer l'effet comique. Ce mécanisme ne met pas réellement en lumière une variation linguistique mais au contraire une utilisation consciente des registres afin de jouer avec la perception de ces derniers chez le lecteur dont la surprise vient de la capacité à identifier les différents registres mis en oeuvre. Ainsi, nous trouvons dans ces exemples un style mimétique de différents registres sciemment utilisés : dès lors, un registre mimé reste il valable ? Ou bien est ce l'illustration de notre perception de ce registre qui est exposé ?

Nous ne visons toutefois pas à répondre à ces questions car notre objectif à terme est de pouvoir évaluer et classer automatiquement une production linguistique, en l'occurrence un texte entier et non un terme ou une phrase isolée seulement. Considérant les registres familier, courant et soutenu, notre but est de construire un jeu de descripteurs (terme que nous définirons plus bas) susceptibles d'être utiles pour cela, c'est-à-dire un ensemble d'éléments quantifiés portant sur des propriétés linguistiques.

3 Présentation de la méthodologie

Nous avons choisi de partir de caractéristiques linguistiques identifiées dans la littérature, que nous avons catégorisées, puis complétées. Nous avons ensuite cherché à valider ou invalider en corpus la nature réellement discriminante de chacune d'entre elles pour les registres considérés. Cette section

présente la méthode de validation, puis le corpus que nous avons construit.

3.1 Validation d'un descripteur

Notre approche de validation s'appuie sur des comparaisons de fréquences d'une caractéristique linguistique entre corpus associés à chaque registre. Pour un corpus donné, chaque caractéristique étudiée (par exemple, l'emploi du mot « ça ») est décrit par sa fréquence d'apparition relative dans un texte, c'est-à-dire normalisée par la longueur en mots du corpus. Par abus de langage, nous abrègerons nos propos en parlant simplement de « descripteur X » pour faire référence à la « fréquence relative d'apparition de la caractéristique X ». Considérant trois corpus textuels, chacun spécifique à un registre, nous posons alors un descripteur comme valide pour un registre donné si, parmi les différents corpus, la valeur du descripteur est significativement [1] supérieure pour le corpus dédié à ce registre à celle des autres.

Cette approche est volontairement simpliste pour rester indépendante d'un maximum d'hypothèses. Notre travail ne prétend ainsi pas statuer de manière absolue sur la validité de tel ou tel descripteur mais dresse un panorama du niveau de fiabilité d'un large panel de descripteurs. Cet étalonnage a pour finalité d'offrir un point de départ et de comparaison à de futurs travaux plus avancés, par exemple des techniques d'analyse plus fines et des motifs extraits automatiquement par des méthodes de fouille. L'approche par fréquence donne en effet différents niveaux de lectures intéressants. D'une part, outre la validité pour un registre, elle permet également de donner la non validité pour un registre dans le cas où un descripteur est significativement inférieur pour ce registre que pour les deux autres. D'autre part, l'analyse contrastive peut être affinée en partitionnant le corpus de chaque registre (Efron, 1979). Ainsi, le nombre de partitions pour lesquelles un descripteur est significativement supérieur permet de quantifier la fiabilité d'un descripteur. Dans cet article, nous présentons des résultats polarisés mais ne reportons pas de niveaux de confiance.

3.2 Corpus

3.2.1 Constitution et traitement du corpus

Les trois types de corpus considérés sont composés d'écrits français : *Albertine disparue* de Proust pour le registre soutenu, des archives de *L'Humanité* et *Le Monde* pour le registre courant et *Kiffe Kiffe Demain* de Guene, *L'Assommoir* de Zola et *Voyage au bout de la nuit* de Céline pour le registre familier. Les trois types de corpus contiennent respectivement $110\,000$, $2\,500\,000$ et $370\,000$ mots. Ces données ont été tokenisées, puis étiquetées en parties du discours avec Treetagger.

3.2.2 Questions soulevées par le corpus

Comme précédemment évoquée, la norme (section 2) peut être envisagée comme le respect des règles grammaticales. Cette première interprétation met en exergue un paradigme où la norme est associée à l'écrit. Par exemple, pour apprendre à maîtriser le français nous faisons des dictées non des improvisations orales : la standardisation interviendrait alors à l'écrit tandis que la variation se

1. S'agissant d'un travail préliminaire, ce critère de significativité n'a pas encore été formalisé.

produirait à l'oral. Ce paradigme sous-tend toute la notion de « registre » puisque l'évaluation opère selon la norme. Ainsi, plus la production semble orale plus elle va être perçue comme familière car spontanée et non structurée, en revanche plus elle paraît écrite plus elle va être appréhendée comme soutenue car perçue plus construite et réfléchie (donc plus rationnelle).

Dès lors, notre corpus est problématique puisque son medium est l'écrit quelque soit son registre. Autrement dit, la parfaite maîtrise des registres par Zola ou Céline, par exemple, invalide-t-il le fait que *L'Assommoir* ou bien *Voyage au bout de la nuit* soient considérés comme du registre familier ?

Toutefois, les travaux présentés se situent au début du projet et ne tendent pas à répondre à ces questions mais à les soulever afin d'explorer plusieurs axes de réflexion.

4 Descripteurs répertoriés

Nous avons dressé une liste de 72 descripteurs, émanant soit de la littérature (66), soit d'une analyse préliminaire conduite par nos soins pour les différents registres (6 descripteurs supplémentaires ont ainsi été identifiés pour le registre familier). Nous sommes conscients de la non exhaustivité de cette liste qui sera amener à évoluer dans de futurs travaux. Ces descripteurs sont généralement soit des motifs « fautifs » (au sens d'un écart à la norme), soit des motifs corrects (toujours selon la norme) mais rares. Ces descripteurs couvrent divers niveaux d'abstraction de la langue que nous regroupons sous les catégories lexicale (16 descripteurs), morphologique (16), syntaxique (38) et phonétique (2).

Dans l'absolu, certains indices lexicaux sont évidemment très discriminants (appartenance explicite de certains lexèmes à un registre donné). Nous n'avons cependant pas traité cet aspect malgré l'important potentiel discriminant de ce type de descripteur : il faudrait disposer de dictionnaires suffisamment exhaustifs [2]. De plus, nous n'avons pas eu recours à une mesure de richesse lexicale car cette notion nous a semblé délicate à traiter. De fait, plus il y a de différents termes lexicaux employés, plus nous pouvons supposer que le vocabulaire est riche donc soutenu. Toutefois, le registre familier est également reconnu pour sa créativité. Pour être efficace, la mesure lexicale devrait ainsi s'appuyer sur une distinction entre termes standards et exotiques, par exemple sur la base d'un dictionnaire à nouveau. Enfin, notons que l'étude de descripteurs phonétiques fait sens y compris pour une analyse de textes écrits (et non transcrits) car l'usage écrit de formes orales est désormais répandu à travers des modes de communications connectés (chats, messageries, textos…).

Quelques exemples sont donnés dans les sous-sections suivantes.

4.1 Phonétique

4.1.1 Elision du « e »

Voyage au bout de la nuit

1. J' vais te dire...
2. J' veux bien payer...
3. J' sais pas encore...

2. Des dictionnaires tels que wiktionnaire par exemple renseignent sur les registres de mots.

4. J’ m’en fous, j’irai me donner.

4.2 Lexical

4.2.1 Sur présence de « là » - ponctuant : familier

Les trois textes associés au registre familier (*L’Assommoir*, *Kiffe Kiffe Demain*, *Voyage au bout de la nuit*) sont ceux où le motif est le plus présent. Suite à une première recherche nous avons voulu préciser la place du motif dans la phrase : un facteur de sa dimension discriminante est sa fonction de « ponctuant ». De fait, lorsque nous avons cherché le motif lorsqu’il était en fin de phrase seuls deux des trois textes catégorisés comme « familiers » (*L’Assommoir*, *Kiffe Kiffe Demain*) ont une fréquence relative plus haute que les autres registres. Il est intéressant de noter que le troisième texte où le motif est le plus présent est du registre soutenu. Nous pourrions, dans des travaux futurs, affiner notre test et rechercher le motif uniquement dans des passages de discours rapporté par exemple.

4.3 Morphosyntaxique

4.3.1 Sujet « nous » transposé en « on »

La présence du sujet « on » est nettement plus présent que « nous » dans les trois oeuvres dont le registre est familier. A l’inverse, c’est dans *Albertine Disparue* associé au registre soutenu que « nous » est le plus présent.

Afin de mieux comprendre le comportement du motif, nous l’avons observé en contexte dans deux oeuvres associées au registre familier.

L’Assommoir :

1. On :
   (a) On la rencontrerait une nuit sur un trottoir, pour sûr.
   (b) Alors, comme on ne parlait pas toujours de leur mariage, elle voulut s’en aller, elle tira légèrement la veste de coupeau.
2. Nous :
   (a) … j’ai quelque chose à laver, je vous garderai une place à côté de moi, et nous causerons.
   (b) — Ca suffit entre nous, madame Gervaise, murmura-t-il.

Kiffe Kiffe Demain :

1. On :
   (a) On lui crie après sans arrêt, et on la surveille pour vérifier qu’elle pique rien dans les chambres.
   (b) Cette meuf, on dirait qu’elle a besoin d’être heureuse à la place des autres.
2. Nous :
   (a) Une fois, il a dit à ma mère qu’en dix ans de métier, c’était la première fois qu’il voyait " des gens comme nous avec un enfant seulement par famille "

(b) Ma mère, elle dit que si mon père nous a abandonnées, c'est parce que c'était écrit.

Ainsi, « on » permet au locuteur de rester impersonnel, ce qui est redoublé par son association à des termes qui généralisent le propos :

1. « on » + « pour sûr »
2. « on » + « toujours »
3. « on » + « sans arrêt »
4. « on » + maxime au présent de vérité général « elle a besoin d'être heureuse à la place des autres »

Tandis que « nous » a tendance à identifier des locuteurs précis :

1. « moi » + vous = « nous »
2. « madame Gervaise » + locuteur = « nous »
3. « ma mère » + « ma » (1 personne du SG) = « nous »
4. « ma mère » + « ma » / « mon » (1 personne du SG = « nous »

4.3.2 Syntagme : « ça + VB »

Lorsque nous observons le comportement du syntagme en contexte, nous constatons que dans les oeuvres associées au registre soutenu et courant, les motifs se trouvent dans des situations de prise de parole explicite avec des marqueurs d'oralité tels que des verbes de parole, les guillemets, la première personne du singulier...

Albertine disparue

1. « (...) si ça amuse le pauvre Swann de faire des bêtises et de ruiner son existence, c'est son affaire, mais on ne se prend pas avec ces choses-là, tout ça peut très mal finir (...) »
2. Et elle ajouta : « Ca devait arriver (...) »

L'Humanité

1. Il pense à « comment ça doit fonctionner ».
2. nous disait : « (...) Et ça va vite, trop vite pour nous.(...) »

Le Monde

1. « D'ordinaire, un conseiller ministériel, petite main e l'ombre, ça ferme sa gueule. » Pierre Jacquemain explique aux « Monde » son départ.
2. le président de la république a défendu ses réformes économiques et martelé que « ça va effectivement mieux pour la France. »

En revanche, dans les oeuvres associées au registre familier, le motif se trouve dans des situations où il n'y a pas une prise de parole explicite.

L'Assommoir

1. Chez elle, ça entrait et ça sortait.
2. Vrai, ça faisait un fameux débarras.

Kiffe Kiffe Demain

1. Ca va rien couter à ta mère si c'est ça qui te préoccupe. De toute facçon, le ski ça pue la merde.
2. Ca marche bien.

Voyage au bout de la nuit

1. Quel effet que ça avait bien pu lui faire ?
2. Ca crève un homme...

Cela met en exergue le lien implicite entre le registre familier et l'oral lorsque ce dernier est représenté à l'écrit. Ce lien pose la question du style plutôt que du genre. En effet, le genre semble être motivé par un facteur extra-linguistique comme le besoin de désambiguiser (faire phonétiquement la différence entre le singulier et le pluriel « il croit » / « ils croivent ») par exemple. Or, ici le besoin semble justement d'être mimétique du genre. Autrement dit, le corpus refléterait l'utilisation de styles pour s'approcher d'un genre. La question de la validité des textes se pose à nouveau. Pour y répondre nous pourrions, dans de futurs travaux, poursuivre notre exploration en utilisant un nouveau corpus oral composé des différents registres.

5 Résultats et discussion

Nous présentons ici les 30 descripteurs ayant été d'ores et déjà validés dans notre corpus, les 42 restants nécessitant soit davantage de ressources textuelles (par extension du corpus initial), soit le recours à des outils plus ou moins complexes (ex. analyseur syntaxique) dont certains sont encore inexistants (par ex., un outil détectant les ellipses, cf. exemple (1), section 2).

La table 1 présente ces 30 descripteurs validés par niveau d'abstraction. Pour chacun, nous indiquons le registre pour lequel il a été validé comme positivement ($+$) ou négativement ($-$) discriminant, ainsi que la référence bibliographique d'où il est tiré. Ceux n'ayant pas de référence sont des descripteurs que nous avons nous mêmes proposés de prendre en considération.

Il ressort que la majorité des descripteurs validés concernent le registre familier. Ce constat s'explique par la créativité de ce registre vis à vis de la norme , qu'il s'agisse de la richesse lexicale puisant dans les lexiques populaires, argotiques voire vulgaires, ou de la multiplicité de manières de s'écarter de la norme (et donc de produire des "fautes"). La notion de faute est d'ailleurs intéressante, non pas pour sa valeur axiologique implicitement associée, mais pour les raisons qui amènent le locuteur à produire un énoncé fautif. Comme le souligne Frei (1929), « on ne fait pas des fautes pour le plaisir de faire des fautes. (...) Dans un grand nombre de cas la faute, qui est passée jusqu'à présent pour un phénomène quasi pathologique sert à prévenir ou à réparer les déficits du langage correct. » Autrement dit, la faute serait le symptôme d'un « déficit » du français. Il y aurait donc une sorte de régularisation spontanée des irrégularités arbitraires de la langue normée. Frei (1929) présente alors les fautes comme venant pallier un « besoin » : besoin de désambiguïser, besoin d'être expressif...

Registre	Source	F	C	S
Niveau lexical (5)				
Éléments ponctuants	(Gadet, 2003)	+		
Onomatopées	(Ilmola, 2012)	+		
« Là » ponctuants	(Gadet, 1997)	+		
Termes à redoublement (« tonton », « dodo »)	(Gadet, 1997)	+		
Planificateurs du discours (« néanmoins », « en raison de »)	(Branca-Rosoff, 1999; Bilger & Cappeau, 2004)			+
Niveau morphosyntaxique (13)				
Contraction de « cela » en « ça »	(Gadet, 1997)	+		−
Négation sans « ne »	(Bilger & Cappeau, 2004)	+		−
Sujet « on » transposé en « nous »	(Bilger & Cappeau, 2004)	−		+
Terminaison en « -asse »	(Ilmola, 2012)	+		
Terminaison en « -ouze »	(Ilmola, 2012)	+		
Terminaison en « -o »	(Ilmola, 2012)	+		
Verbe « être » au singulier devant un syntagme nominal singulier	(Bilger & Cappeau, 2004; Favart, 2011)	+		
« Ça » + verbe		+		
Dérivation en adverbe d'un nom ou adjectif (« vachement »…)	(Ilmola, 2012)	+		
Verbes du premier groupe	(Gadet, 1997)	+		
Emploi du passé simple		−		+
Emploi du passé composé			+	−
Emploi du présent de l'indicatif			+	
Niveau syntaxique (10)				
Emploi fautif de relatives en « que » (« relative populaire »)	(Gadet, 2003)	+		
Interrogative sans inversion sujet/verbe	(Gadet, 2003)	+		
Interrogative en « est-ce que »	(Ilmola, 2012)		+	
Maintien de « des » devant un adjectif au lieu de « de »	(Ilmola, 2012; Kalmbach, 2012)	+		
Rajout de « à lui/elle » après un pronom personnel « son/sa »	(Gadet, 2003)	+		
Emploi de pronoms relatifs (« dont », « lequel »…)	(Gadet, 2003)			+
Adverbe + parataxe (« vraiment bien »…)		+		
Inversion « en » et COI à l'impératif (« donne m'en »)	(Kalmbach, 2012)	+		
« C'est …qui » (« c'est lui qui a fait ça »)		+		
Effacement du pronom « il » impersonnel (« fallait pas… »)	(Favart, 2011)	+		
Niveau phonétique (2)				
Élision de « e » (Favart, 2011)		+		
Élision du « i » du pronom « qui » devant une voyelle	(Ilmola, 2012)	+		

TABLE 1 – Descripteurs validés positivement (+) et négativement (−) pour les registres familier (F), courant (C) et soutenu (S).

Il arrive que des fréquences relatives d'un même motif dans deux textes associés à des registres différents soient très proches. C'est le cas par exemple des « termes à redoublement ». On observe qu'ils ont la fréquence relative la plus importante dans Kiffe Kiffe Demain (associé au registre familier) mais que la seconde fréquence relative la plus haute se trouve dans Albertine Disparue (associé au registre soutenu). Dans ce cas précis, nous avons décidé malgré tout de valider ce motif en tant qu'évocateur du registre familier mais ceci a attiré notre attention et nous avons alors observé plus en détail le contexte du motif avec un concordancier. Nous avons notamment alors remarqué une corrélation entre la présence de discours rapportés (identifiés via la présence de verbes de parole ou de marques de ponctuation telles que les guillemets) et les descripteurs discriminants pour le registre familier, ceci semblant étayer le fait (qu'il faudrait bien entendu approfondir) qu'une certaine oralité serait plus étroitement associée au registre familier. Nous avons par ailleurs constaté que l'apparition d'un descripteur familier était souvent doublée par la présence d'autres descripteurs familiers : « c'est ça » (forme contractée + redondance sémantique : « cela est cela »), « ça rime à quoi ? » (forme contractée + non inversion de forme interrogative + phrase courte). Ainsi, un descripteur serait toujours renforcé par la présence d'un second descripteur. Nous pourrions donc, dans de futurs travaux, chercher à identifier des n-uplets de descripteurs plutôt que des descripteurs isolés.

D'un point de vue théorique, il est clair que nous nous plaçons dans une étude qui rend centrale – conjointement à celle de l'écart à la norme – la question du continuum oral-écrit, à l'instar des travaux de (Blanche-Benveniste, 1997), lesquels ont notamment souligné que ce qui est souvent présenté comme spécifique des modalités orale *vs.* écrite concerne (aussi) l'opposition entre registres formel *vs.* informel. Dans une méthodologie proche de celle de (Douglas, 1988), nos travaux devraient permettre de proposer un ensemble exhaustif - et nous l'espérons pertinent - de descripteurs linguistiques à même d'éclairer ces deux types d'oppositions.

6 Conclusion

Dans le travail présenté ici, nous sommes revenus sur des caractéristiques identifiées dans la littérature (au nombre de 72) abordant la notion de registre pour en proposer un mode de validation puis une catégorisation utile pour des tâches futures auxquelles nous allons nous atteler prochainement (classification, extraction de motifs discriminants). Le résultat de ce travail est ainsi une première liste de 30 descripteurs validés parmi 48 descripteurs testés.

Outre le fait de continuer l'investigation des 24 descripteurs restants non encore testés, un prolongement immédiat à ce travail concerne l'établissement de degrés de confiance associés aux descripteurs validés, par exemple par des techniques de *bagging* et d'analyse de la variance. Ce travail devrait également concerner les 18 descripteurs que nous avons testés mais que nos corpus n'ont pas permis de valider de manière certaine. De manière corrélée, un second axe de travail contribuant également à une vision plus fine des phénomènes étudiés est la constitution d'un corpus de grande ampleur et faisant intervenir de manière croisée et explicite tout à la fois les notions de genre (en nous appuyant sur les réflexions de Adam (1999)) et de registre (dans le sens où nous l'avons défini en section 2). Cette première liste de descripteurs validés oeuvre en ce sens car elle devrait permettre de construire un premier outil (règles expertes, classifieur simple) pour filtrer et annoter automatiquement de grands corpus (libres, journaux, pages web…).

Remerciements

Ce travail a bénéficié du soutien financier de l'Agence Nationale de la Recherche (ANR) dans le cadre du projet TREMoLo (ANR-16-CE23-0019).

Références

ADAM J.-M. (1999). *Linguistique textuelle. Des genres de discours aux textes*. Nathan.

BELL A. (1984). Language style as audience design. *Language in society*, **13**(2), 145–204.

BERGSON H. (2013). *Le rire*. Flammarion.

BIBER D. & FINEGAN E. (1994). *Sociolinguistic perspectives on register*. Oxford University Press on Demand.

BILGER M. & CAPPEAU P. (2004). L'oral ou la multiplication des styles. *Langage et société*, (3).

BLANCHE-BENVENISTE C. (1997). Approches de la langue parlée en français. *Ophrys, Paris*.

BORZEIX A. & FRAENKEL B. (2005). *Langage et travail (communication, cognition, action)*. CNRS éd.

BRANCA-ROSOFF S. (1999). Des innovations et des fonctionnements de langue rapportés à des genres. *Langage et société*, **87**(1), 115–129.

DOUGLAS B. (1988). Variation across speech and writing. *Cambridge : CUP*.

FAVART F. (2011). Le stéréotype de registre de langue populaire dans le roman du second XXe siècle (1966-2006). In *Stéréotypes en langue et en discours*. Centre Interlangues.

FERGUSON C. A. (1982). Simplified registers and linguistic theory. *Exceptional language and linguistics*, p. 49–66.

FREI H. (1929). *La grammaire des fautes : introduction à la linguistique fonctionnelle, assimilation et différenciation, brièveté et invariabilité, expressivité*, volume 1. Slatkine.

GADET F. (1996). Variabilité, variation, variété : le français d'europe. *Journal of French Language Studies*, **6**(1), 75–98.

GADET F. (1997). La variation, plus qu'une écume. *Langue française*, p. 5–18.

GADET F. (2003). Is there a french theory of variation ? *International Journal of the Sociology of Language*, **165**.

ILMOLA M. (2012). Les registres familier, populaire et vulgaire dans le canard enchaîné et charlie hebdo : étude comparative.

KALMBACH J.-M. (2012). *La grammaire du français langue étrangère pour étudiants finnophones*. Kielten laitos, Jyväskylän yliopisto.

LEDEGEN G. & LÉGLISE I. (2013). *Variations et changements linguistiques*. ENS Editions.

SANDERS C. (1993). *Sociosituational variation*. Cambridge : Cambridge University Press.

URE J. (1982). Introduction : approaches to the study of register range. *International Journal of the Sociology of Language*, **1982**(35), 5–24.

Construction d'un corpus multilingue annoté en relations de traduction

Yuming Zhai
LIMSI/CNRS, Univ. Paris-Sud, Université Paris-Saclay, 91405 Orsay
yuming.zhai@limsi.fr

RÉSUMÉ

Les relations de traduction, qui distinguent la traduction littérale d'autres procédés, constituent un sujet d'étude important pour les traducteurs humains (Chuquet & Paillard, 1989). Or les traitements automatiques fondés sur des relations entre langues, tels que la traduction automatique ou la méthode de génération de paraphrases par équivalence de traduction, ne les ont pas exploitées explicitement jusqu'à présent. Dans ce travail, nous présentons une catégorisation des relations de traduction et nous les annotons dans un corpus parallèle multilingue (anglais, français, chinois) de présentations orales, les *TED Talks*. Notre objectif à plus long terme sera d'en faire la détection de manière automatique afin de pouvoir les intégrer comme caractéristiques importantes pour la recherche de segments monolingues en relation d'équivalence (paraphrases) ou d'implication. Le corpus annoté résultant de notre travail sera mis à disposition de la communauté.

ABSTRACT

Construction of a multilingual corpus annotated with translation relations

Translation relations, which distinguish literal translation from other translation techniques, constitute an important subject of study for human translators (Chuquet & Paillard, 1989). However, automatic processing techniques based on interlingual relations, such as machine translation or paraphrase generation exploiting translation equivalence, have not exploited these relations explicitly until now. In this work, we present a categorisation of translation relations and annotate them in a parallel multilingual (English, French, Chinese) corpus of oral presentations, the TED Talks. Our long term objective will be to automatically detect these relations in order to integrate them as important characteristics for the search of monolingual segments in relation of equivalence (paraphrases) or of entailment. The annotated corpus resulting from our work will be made available to the community.

MOTS-CLÉS : relations de traduction, annotation de corpus.

KEYWORDS: translation relations, corpus annotation.

1 Introduction

Dans le domaine des recherches en acquisition automatique de paraphrases, les premiers travaux guidés par les données ont exploité des corpus monolingues. Les méthodes proposées ont notamment reposé sur l'analyse des contextes environnants (Barzilay & McKeown, 2001), des calculs de similarité fondé sur des arbres de dépendances (Lin & Pantel, 2001; Ibrahim *et al.*, 2003), la fusion d'arbres de constituants (Pang *et al.*, 2003), ou le regroupement de documents par des critères de dates et de thèmes (Dolan *et al.*, 2004).

Une autre famille d'approches importante exploite des corpus multilingues parallèles, disponibles en abondance pour certaines paires de langues et certains domaines. L'approche la plus étudiée repose sur l'équivalence de traduction entre segments (Bannard & Callison-Burch, 2005), et sur l'hypothèse selon laquelle si deux segments dans la même langue partagent une ou plusieurs traductions communes (considérées comme des "pivots") dans une ou plusieurs langues étrangères, alors ils sont potentiellement des paraphrases (voir une illustration sur la figure 1). Cette méthode exploite les informations des tables de traduction statique générées par les systèmes de traduction automatique basés sur les segments (PBSMT). Le travail ultérieur de Callison-Burch (2008) a affiné cette approche en imposant que les segments partagent la même structure syntaxique *CCG (Combinatory Categorial Grammar)*, ce qui a permis d'améliorer la substituabilité grammaticale pour les paires produites. En se basant sur cette même approche, mais dans le but d'obtenir une meilleure généralisation, Zhao *et al.* (2008) ont utilisé des arbres de dépendances pour apprendre des patrons de paraphrases qui incluent des variables de partie du discours.

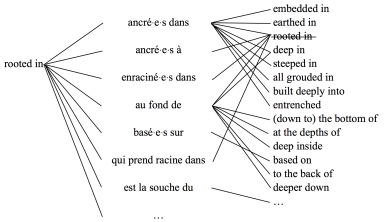


FIGURE 1: Exemple de génération de paraphrases par pivot en français pour le segment anglais «rooted in».

La méthode dite «par pivot» a été mise en œuvre pour la construction de la ressource PPDB (Paraphrase Database) [1], aujourd'hui la plus grande ressource de paraphrases disponible avec plus de 100 millions de paires de segments anglais (Ganitkevitch *et al.*, 2013). La construction de cette ressource a été rendue possible par l'utilisation d'un corpus parallèle de plus de 106 millions de paires de phrases (soit plus de 2 milliards de mots anglais) couvrant 22 langues pivot. La version multilingue de PPDB (Ganitkevitch & Callison-Burch, 2014) contient des paires de segments pour 23 langues, dont le français, obtenues en utilisant l'anglais comme langue pivot. Chaque paire de segments est associée à une trentaine de caractéristiques, notamment la probabilité de paraphrase (Bannard & Callison-Burch, 2005) et des scores de similarité distributionnelle monolingue (Ganitkevitch *et al.*, 2012). De plus, chaque paire partage une même catégorie grammaticale selon les contraintes imposées dans (Callison-Burch, 2008). Le score de classement dans la version initiale de PPDB est fondé sur un calcul combinant un sous-ensemble de caractéristiques avec des pondérations *ad hoc* fondées sur les intuitions des auteurs.

Pour la seconde version de cette ressource, PPDB 2.0 (Pavlick *et al.*, 2015b), un modèle de régression a été utilisé afin d'adapter le score de paraphrase à des jugements humains de la qualité des paraphrases, permettant une meilleure corrélation qu'avec le classement heuristique de PPDB 1.0. De manière importante, le travail de Pavlick *et al.* (2015a) a mis en évidence le fait qu'il existe d'autres relations sémantiques que l'équivalence stricte (paraphrase) dans une telle ressource obtenue par l'équivalence de traduction. Ce travail décrit une catégorisation automatique de diverses relations (*Équivalence, Implication (dans les deux sens), Exclusion, Autrement lié et Indépendant*) ayant ex-

1. http://paraphrase.org

ploité de nombreuses caractéristiques incluant : des informations de niveau lexical, des informations issues de WordNet (Miller, 1995), des patrons lexico-syntaxiques, des valeurs de similarité distributionnelle entre vecteurs de contexte de dépendances, des probabilités de paraphrase, le nombre total de traductions partagées pour chaque paire de phrases, etc. La meilleure combinaison, qui utilise à la fois des caractéristiques monolingues et des caractéristiques bilingues, permet d'atteindre une précision globale de 79%. Une estimation réalisée sur la plus grande taille de PPDB montre qu'il existerait tout au plus seulement 10% de paraphrases strictes. Nous pouvons donc en conclure qu'une meilleure représentation sémantique est nécessaire pour améliorer cette technique, que ce soit pour obtenir des paraphrases ou pour obtenir de manière contrôlée d'autres types de variantes.

La traduction automatique, qui repose elle aussi sur des correspondances bilingues, a connu récemment des améliorations significatives avec l'avènement des techniques neuronales (NMT), permettant pour des langues proches d'atteindre des performances plus proches de traductions humaines que par les techniques statistiques (SMT) précédentes (Wu *et al.*, 2016). Le travail de Lapata *et al.* (2017) a consisté à réimplémenter la méthode de génération de paraphrases par pivot en utilisant des systèmes NMT. Les résultats expérimentaux obtenus dans le cadre de plusieurs tâches (prédiction de la similarité, identification des paraphrases et génération de paraphrases) montrent que leur approche améliore les approches précédentes.

Ces travaux en génération de paraphrases et en traduction automatique n'ont cependant, à notre connaissance, jamais pris en compte les relations de traduction entre paires de segments, alors que cela correspond à un sujet très important pour les traducteurs humains. Les cas de traduction *littérale* sont bien exploités par l'utilisation de grands corpus et par les systèmes neuronaux. En revanche, il existe un très grand nombre de traductions non littérales, en particulier dans des genres textuels non techniques. Ces traductions posent souvent des difficultés pour l'alignement de mots, essentiel pour les méthodes statistiques, et elles peuvent faire dévier le sens originel du texte source. Le manque de modélisation de ces relations conduit à une perte de contrôle sur la sémantique produite, ce qui est attesté par les diverses relations sémantiques dans la ressource PPDB (Pavlick *et al.*, 2015a). De plus, les différences culturelles donneront éventuellement des distributions de relations de traduction différentes en fonction des langues. Pour la génération de variantes par équivalence de traduction, des langues pivots différentes peuvent éventuellement produire des résultats très différents, ce qui n'a pas été considéré jusque-là.

Dans cet article, nous catégorisons ces relations de traduction en modélisant le choix des traducteurs humains qui les ont produites, et nous les annotons dans un corpus multilingue de discours préparés, les *TED Talks* [2]. Nous décrivons les définitions des relations, le processus d'annotation ainsi que les principaux problèmes rencontrés. Notre objectif suivant portera sur la détection automatique de ces relations afin de les intégrer comme caractéristiques dans la recherche de paraphrases ou de paires en relation d'implication. Nous faisons l'hypothèse que cela permettra davantage de contrôle sémantique et de variété en génération. Nous présentons les travaux précédents en lien avec notre travail dans la Section 2, et décrivons notre corpus dans la Section 3. Les relations de traduction sont décrites dans la Section 4, suivies par le processus et les statistiques des annotations obtenues dans la Section 5. Nous donnons finalement nos conclusions et perspectives dans la Section 6.

2. https://www.ted.com/

2 Travaux précédents

Le travail de Deng & Xue (2017) a étudié les divergences présentes dans la traduction automatique anglais-chinois à l'aide d'un schéma d'alignement hiérarchique entre des arbres d'analyse pour ces deux langues. Sept types de divergences ont été identifiées, certaines posant des difficultés importantes pour l'alignement automatique de mots, notamment les différences lexicales résultant de traductions non littérales, et les différences de structures entre langues (avec ou sans changement de types de syntagmes). En vue de fournir un jeu de données particulier sur les expressions multi-mots pour la traduction automatique, Monti *et al.* (2015) ont annoté spécifiquement ces expressions dans le corpus *TED Talks* anglais-italien associées à leur traduction générée par un système automatique. Les phénomènes discutés dans les deux travaux que nous venons de mentionner sont inclus dans les relations de traduction que nous présentons dans ce travail.

Reprenant l'approche de génération de paraphrases par pivot, Kok & Brockett (2010) ont introduit un modèle à base de graphes présenté sous le nom de *HTP (Hitting Time Paraphraser)*. Cette approche repose sur des parcours aléatoires et sur le temps d'atteinte (*hitting time*) afin d'extraire des paraphrases à partir de corpus parallèles multilingues. Cette approche parcourt des chemins de longueur supérieure à 2 en utilisant l'information entre les nœuds représentant des segments dans une autre langue et en permettant d'intégrer des connaissances monolingues sous forme de nœuds spéciaux. Les résultats expérimentaux ont permis d'obtenir davantage de paraphrases correctes que l'approche de Callison-Burch (2008) [3] ainsi qu'une meilleure précision pour les paraphrases classées aux premiers rangs. Nous comptons par la suite poursuivre ce travail en nous intéressant spécifiquement aux relations de traduction non-littérales afin d'étudier si un changement de sens a lieu, dans l'espoir de mieux guider le parcours dans des corpus multilingues pour obtenir des paraphrases par équivalence de traduction.

Une limite importante de l'approche par pivot est qu'elle ne distingue pas les différents sens possibles d'un segment lors de la génération de ses paraphrases potentielles. Le travail de Apidianaki *et al.* (2014) a analysé la sémantique des paraphrases lexicales obtenues avec l'approche de Callison-Burch (2008) et a mis en évidence la nécessité d'une étape de désambiguïsation. Le travail ultérieur de Cocos & Callison-Burch (2016) a introduit une méthode pour effectuer le regroupement des paraphrases par sens, laquelle a été appliquée à la ressource PPDB. Le travail que nous présentons dans cet article porte davantage sur les relations de traduction (bilingue) qui sont à l'origine des diversités sémantiques dans une ressource telle que PPDB, ainsi que sur leur exploitation pour la suite de notre travail. La prise en compte de la polysémie nous concernera dans un second temps.

3 Corpus

Afin de faire l'étude des relations de traduction pour plusieurs paires de langues, nous avons travaillé sur un corpus parallèle multilingue. Le corpus annoté est issu de l'inventaire Web *WIT*^{3} (Cettolo *et al.*, 2012) qui donne accès à une collection de conférences transcrites et traduites incluant le corpus *TED Talks* [4]. Ce corpus a été mis à disposition pour les campagnes d'évaluation IWSLT 2013 et 2014 [5]. La

3. La méthode d'évaluation est toutefois une version simplifiée de celle utilisée dans (Callison-Burch, 2008).

4. `https://wit3.fbk.eu/`

5. Nous avons utilisé le corpus d'entraînement de 2014 (160 656 lignes), de développement (880 lignes) et de test (1 556 lignes) de 2010.

langue d'origine, c'est-à-dire dans laquelle se sont originellement exprimés les orateurs, est l'anglais. Nous avons calculé l'intersection des corpus parallèles avec les traductions en français [6], chinois, arabe, espagnol et russe. La traduction des sous-titres de *TED Talks* est contrôlée par des bénévoles et des coordinateurs par langue [7], permettant une traduction d'un bon niveau de qualité en général. Le corpus annoté contient 2 436 lignes de phrases parallèles pour chaque paire de langues. À ce stade, nous décrivons le début de nos annotations pour les paires anglais-français et anglais-chinois. La table 3 décrit les statistiques principales des corpus correspondants.

Pour l'anglais et le français, la tokenisation est réalisée par l'outil Stanford Tokenizer [8]. Les lettres capitales au début de chaque ligne ont été transformées en minuscules, si et seulement si les mots en question ont par ailleurs leur première lettre en minuscule dans le corpus. Dans le cas contraire, les lettres capitales sont gardées telles quelles pour les mots qui apparaissent toujours avec des initiales en majuscule. Nous avons utilisé l'outil THULAC (Li & Sun, 2009) pour la segmentation du corpus chinois. L'alignement automatique de mots des corpus bilingues a été réalisé par l'outil FastAlign (Dyer *et al.*, 2013) avec ses paramètres par défaut et en l'entraînant sur l'intégralité de chaque corpus parallèle (soit 163 092 lignes et 3 303 660 tokens anglais).

4 Relations de traduction

Nous avons établi une hiérarchie décrivant les relations de traduction en nous fondant sur les théories explicitées dans l'ouvrage de Chuquet & Paillard (1989) et les phénomènes rencontrés pendant notre étude de corpus initiale (voir figure 2). Les nœuds colorés représentent nos catégories, les

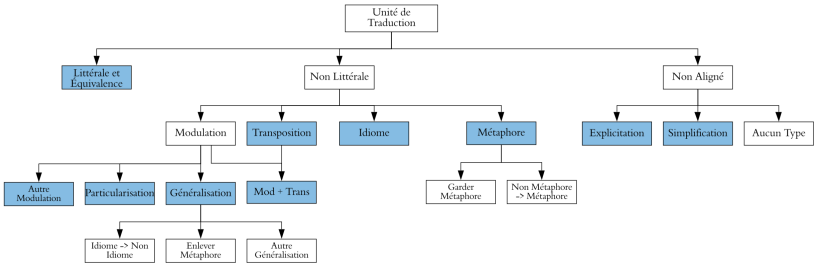


FIGURE 2: Hiérarchie de relations de traduction.

autres nœuds décrivant la hiérarchie (*i.e. Non Littérale, Non Aligné, Modulation, Aucun Type*) ou des phénomènes plus précis mais pour lesquels une étiquette dédiée n'a pas été retenue (*e.g. Enlever Métaphore, Autre Généralisation*). Nous présentons ci-dessous leur définition ainsi que des exemples caractéristiques :

1. Traduction littérale : traduction mot à mot, ce qui concerne également les situations où des idiomes peuvent être traduits de façon littérale :

   facts are stubborn -> les faits sont têtus, *What time is it ? -> Quelle heure est-il ?*

6. Les frontières de phrases ont été corrigées dans le corpus de test français pour calculer l'intersection.
7. `https://www.ted.com/participate/translate/get-started`
8. `http://nlp.stanford.edu/software/tokenizer.shtml`

2. Équivalence :

   i) Traduction non-littérale de certains proverbes, idiomes ou expressions figées

   Birds of a feather flock together. -> Qui se ressemble s'assemble.

   on the brink of -> à deux doigts de

   ii) Équivalence sémantique au niveau supra-lexical, ou traduction des termes

   magic trick -> tour de magie, *hatpin -> épingle à chapeau*

3. Généralisation : cette catégorie inclut trois sous-types, mais nous les annotons par une seule étiquette :

   i) La traduction est plus générale ou neutre ; dans d'autres cas, ce procédé rend le sens plus accessible dans la langue cible :

   look carefully at -> regardez, *as we **sit here** in ... -> alors que nous **sommes** à ...*

   ii) Traduction d'un idiome par une expression non figée :

   trial and error -> procéder par tâtonnements

   iii) Suppression d'une image métaphorique :

   *ancient Tairona civilization which once **carpeted** the Caribbean coastal plain -> anciennes civilisations tyranniques qui **occupaient** jadis la plaine côtière des Caraïbes*

4. Particularisation : la traduction est plus précise ou présente un sens plus concret :

   *the director **said** -> le directeur **déclara***, *language loss -> l' extinction du langage*

5. Modulation : ce procédé consiste à changer le point de vue, soit pour contourner une difficulté de traduction, soit pour révéler une manière différente de voir les choses pour les locuteurs de la langue cible :

   this is a completely unsustainable pattern -> il est absolument impossible de continuer sur cette tendance, *I had an assignment -> on m'avait confié une mission*

6. Transposition : traduction des mots ou des expressions à l'aide d'autres catégories grammaticales que celles utilisées dans la langue source, sans pour autant modifier le sens de l'énoncé :

   astonishingly inquisitive -> dotée d'une curiosité stupéfiante

   *patients **over** the age of 40 -> les malades **ayant dépassé** l'âge de 40 ans*

7. Modulation plus Transposition : ce type peut contenir n'importe quel sous-type de modulation combiné avec la transposition :

   *this is a people **who cognitively do not distinguish** -> c'est un peuple **dont l'état des connaissances ne permet pas de faire la distinction***

8. Idiome : cas de la traduction d'expressions non figées par un idiome (très fréquent dans la traduction de l'anglais en chinois) :

   at any given moment -> à un instant "t"

   died getting old -> 行将就木(getting closer and closer to the coffin)

9. Métaphore : cette catégorie inclut deux sous-types ramenés à une seule étiquette :

   i) Conservation d'une métaphore à l'aide d'une traduction non littérale :

   *the Sun begins to **bathe** the slopes of the landscape -> le soleil qui **inonde** les flancs de ce paysage*

   ii) Introduction d'une métaphore pour traduire des segments non métaphoriques :

   *if you **faint** easily -> si vous **tombez dans les pommes** facilement*

10. Non aligné - Explicitation : introduction dans la langue cible de clarifications pour des éléments implicites dans la langue source mais qui émergent du contexte ou de la situation :
    *feel their past in the wind -> ressentent leur passé **souffler** dans le vent*
11. Non aligné - Simplification : non traduction délibérée de certains mots pleins :
    *and you'll **suddenly** discover what it would be like -> et vous découvrirez ce que ce serait*
12. Non aligné et aucun type attribué : mots outils nécessaires dans une langue mais pas dans l'autre ; segments non traduits mais qui n'influencent pas le sens ; segments donnant des informations répétées en contexte :
    *minus 271 degrees, colder than -> moins 271 degrés, **ce qui est** plus froid*
    *the last example I have time to -> le dernier exemple **que** j'ai le temps de*

5 Annotation des relations

Outil et configuration Nous avons utilisé l'application Web Yawat [9] (Germann, 2008), qui nous permet d'aligner des mots ou des segments (continus ou discontinus), puis d'attribuer des étiquettes configurables adaptées pour notre tâche à des unités monolingues ou bilingues (voir figure 3) .

À des fins d'illustration, considérons l'exemple trilingue suivant :

well, we use that great euphemism, "trial and error", which is exposed to be meaningless.
eh bien, nous employons cet euphémisme, procédé par tâtonnements, qui est dénué de sens.
我们(nous) 普通人(les gens ordinaires) 会(particule du temps futur) 做(faire) 各种各样(divers) 的(particule pour attribut) 实验(expérience) 不断(sans arrêt) 地(particule pour adverbe) 犯错误(commettre une faute) 结果(par conséquent) 却(cependant) 一无所获 (ne rien gagner)

Les segments *well* et *we use that great euphemism* sont traduits littéralement en français mais sont omis en chinois. L'idiome *trial and error* est traduit par une généralisation dans les deux langues. Le segment *which is exposed to be* est traduit par une généralisation en français (*est*) et par une modulation en chinois (*结果(par conséquent) 却(cependant)*). L'adjectif *meaningless* est traduit par une transposition en français (*dénué de sens*) et par un idiome de quatre caractères en chinois (*一无所获(ne rien gagner)*).

Le corpus ayant été préalablement aligné automatiquement, nous avons importé les alignements produits en vue d'accélérer le processus d'alignement, en particulier pour les mots traduits littéralement. Les annotateurs avaient pour consigne de corriger ces alignements si nécessaire. Le corpus chinois

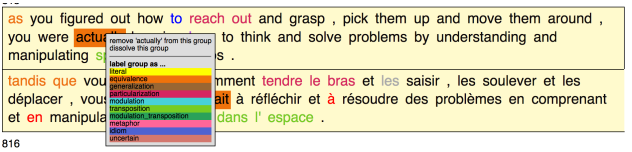


FIGURE 3: Interface de l'outil Yawat permettant d'effectuer l'annotation.

segmenté automatiquement contient des erreurs qui peuvent produire des alignements incorrects puis

9. *Yet Another Word Alignment Tool* ; cet outil est disponible pour la recherche sous la licence GNU Affero General Public License v3.0.

des attributions d'étiquette incorrectes. Certains mots chinois ont donc été resegmentés manuellement préalablement à l'annotation afin de mieux correspondre aux segments anglais (par exemple, *only is* -> 仅仅是 a été corrigé en : *only is* -> 仅仅*(only)* 是*(is)*).

Les éventuelles fautes d'orthographe dans le corpus originel n'ont, par contre, pas été corrigées, car d'une part il n'en existe pas beaucoup et d'autre part cela n'empêche pas l'attribution de catégories en général. Toutefois, nous avons introduit une catégorie *Incertain* pour des paires pour lesquelles les annotateurs ne savent pas attribuer de catégorie ou pour des paires qui contiennent des erreurs de traduction manifestes.

Deux annotateurs [10] ont participé à ce travail préliminaire pour la paire anglais-français, et une seule annotatrice pour la paire anglais-chinois. La formation des futurs annotateurs s'appuiera sur un guide d'annotation qui définit l'ensemble des types illustrés par des exemples. La hiérarchie des catégories permet de donner une vue d'ensemble des relations entre les catégories, et des exemples discriminants permettent de mieux guider les annotateurs dans les étapes de décision. En vue de bien comprendre le contexte, les annotateurs peuvent regarder des vidéos de conférences [11] correspondantes avant d'annoter.

Étude de contrôle Nous avons évalué de manière conventionnelle la faisabilité de notre tâche d'annotation en mesurant un accord inter-annotateurs sur un corpus de contrôle. Deux annotateurs ont annoté indépendamment 100 paires de phrases (3 055 tokens anglais et 3 238 tokens français). Puisqu'il existe des désaccords sur la frontière de certains segments, nous avons calculé la valeur du Kappa de Cohen (Cohen, 1960) uniquement pour les segments de mêmes frontières et obtenu la valeur 0,672, qui signifie un accord fort. Le nombre de tokens anglais annotés dans des segments de mêmes frontières est de 1 906 pour la catégorie *Littérale* et de 312 pour les autres catégories, ce qui couvre 72,60% des tokens source anglais. Si nous calculons un accord inter-annotateur de manière plus flexible en incluant des paires avec des segmentations différentes mais compatibles (i.e. pas de chevauchement aux frontières) mais avec une annotation commune [12], la valeur de Kappa diminue à 0,617, ce qui correspond à un accord fort-modéré. Cependant, dans cette configuration, la couverture des tokens anglais augmente à 85,56%. Les tokens restant appartiennent eux à des segments aux frontières incompatibles.

Nous présentons la table de confusion pour ces segments (sans considérer les appariements flexibles) dans la figure 4. Sans surprise, la majorité des situations d'accord correspondent aux traductions littérales. Il existe tout de même certains désaccords pour ce type avec *Équivalence (e.g. in this way -> de cette façon), Modulation (e.g. this entire time -> tout ce temps), Particularisation (e.g. snuff -> tabac)* et *Transposition (e.g. their prayers* ***alone*** *->* ***seulement*** *leurs prières)*. Néanmoins nos catégories *Équivalence* et *Littérale* sont très proches, et *Particularisation* est un sous-type de *Modulation*, confusions que nous aurions donc pu considérer comme admissibles dans une mesure plus flexible. *Modulation* présente le plus grand nombre de confusions avec *Littérale* et *Transposition (e.g. from the forest floor -> tombées par terre)*, ce qui indique qu'il est nécessaire de mieux expliciter leurs différences quand nous formons les annotateurs. *Mod+Trans* est un type combiné pour lequel certains annotateurs ne perçoivent parfois que l'un des deux types (*e.g. a great distance -> de loin*). Il existe très peu de confusions pour *Généralisation (e.g. because they're denatured -> étant dénaturés)* mais c'est moins le cas pour *Particularisation*. *Non aligné - Explicitation* et *Non aligné -*

10. Un annotateur français et une annotatrice chinoise.

11. Les corpus à annoter sont des transcriptions des *TED Talks* et leur traduction.

12. Par exemple, *I was asked by* et *I was asked by my professor at Havard* ont tous les deux été annotés par *Modulation*, et *my professor at Havard* a été annoté par *Littérale* par le premier annotateur. Nous considérons ici que les deux segmentations sont compatibles et qu'il y a accord entre les deux annotateurs sur un segment (le plus grand) du fait du type commun *Modulation*.

Simplification présentent très peu de confusions avec les autres types, mais sont parfois en compétition avec *Aucun Type*. *Métaphore* est à l'origine de quelques désaccords (*e.g. at the **base** of glaciers -> aux **pieds** des glaciers*), qui peuvent notamment s'expliquer par la difficulté d'annotation pour un annotateur non natif de la langue cible.

	Équivalence	Littérale	Modulation	Transposit.	Mod+Trans	Généralisat.	Particulari.	Explicitation	Simplificat.	Idiome	Métaphore	Incertain
Équivalence	21	4	0	5	0	0	1	0	1	0	0	0
Littérale	27	1857	26	6	0	0	10	0	0	0	0	7
Modulation	4	8	37	7	1	1	3	0	2	0	0	0
Transposition	6	7	10	30	0	1	0	0	0	0	0	0
Mod+Trans	0	1	6	2	2	2	0	0	0	0	0	0
Généralisation	0	1	0	0	0	17	0	0	0	0	0	1
Particularisation	4	13	6	2	0	1	29	0	0	0	0	2
Explicitation	0	0	0	0	0	0	0	10	0	0	0	0
Simplification	0	0	0	0	0	0	0	0	40	0	0	0
Idiome	0	0	0	0	0	0	0	0	0	0	0	0
Métaphore	1	0	0	0	0	1	1	0	0	0	1	0
Incertain	1	8	2	2	0	4	0	0	1	0	0	4

FIGURE 4: Table de confusion pour le corpus de contrôle (nombre d'instances).

Processus à plusieurs passes Le calcul d'une valeur d'accord inter-annotateurs permet certaines interprétations standard pour le corpus de contrôle, et la table de confusion nous aide à identifier des difficultés de la tâche. Afin de converger sur les frontières de segments et sur les attributions de types, nous avons adopté un processus d'annotation à plusieurs passes en vue d'obtenir une meilleure qualité d'annotation. Pour chaque sous-corpus [13], un premier annotateur réalise une première passe pour l'ensemble des catégories puis un deuxième annotateur prend le relais, ce qui lui permet de modifier les alignements et/ou les catégories s'il existe un désaccord. Chaque fichier d'annotation est sauvegardé à l'issue de chaque passe pour documenter les différences dans l'annotation. Cette alternance peut se répéter jusqu'à la convergence de toutes les annotations. En pratique, nous nous limitons à 3 passes, la 3ème étant effectuée par le premier annotateur du corpus. Nous constatons que le nombre de modifications dans la 3ème passe décroît au fur et à mesure des documents annotés, rendant compte d'une adaptation progressive et rapide des annotateurs à la tâche. Ce mode d'annotation, plus coûteux, a toutefois été rendu nécessaire par la qualité visée ainsi que par les difficultés inhérentes à la segmentation observées sur le corpus de contrôle.

Certaines situations nécessitent l'établissement de conventions d'annotation pour garantir la cohérence des annotations (voir table 1). Par exemple, pour des articles français n'ayant pas de correspondance en anglais, nous attachons ceux-ci avec le nom modifié pour préciser leur appartenance, *e.g. play with **blocks** -> jouer avec **des cubes***. Tout changement en nombre, temps de verbe (sauf si rendu nécessaire par le contexte), pronom personnel, préposition (traduction non littérale) et ponctuation est considéré comme *Modulation*.

Comme nous l'avons vu, les frontières de segments peuvent différer selon les annotateurs. La procédure à plusieurs passes par alternance permet de faire disparaître progressivement ce type de désaccords. Par exemple : *a learning tool for **language learners** -> un outil d'apprentissage pour **ceux qui apprennent des langues***, le premier annotateur avait séparé *language* et *learners*, le second les avait ensuite regroupés en attribuant le type *Modulation+Transposition*, ce qui a été finalement approuvé par le premier annotateur en troisième passe. Cet autre exemple consiste en une séparation : ***I want to start by** showing you -> je **vais** vous montrer* ; les deux annotateurs sont finalement tombés d'accord pour ne pas inclure *showing* et *montrer* dans la paire de type *Généralisation* : *want to start by -> vais*.

13. Chaque sous-corpus représente une ou plusieurs interventions complètes aux *TED Talks*, afin de mieux comprendre le contexte.

Pour des types *Équivalence*, *Particularisation* et *Métaphore*, des annotateurs natifs de la langue cible sont plus à l'aise pour prendre des décisions. Quand un annotateur hésite sur le choix d'une catégorie appropriée, une bonne pratique est de réfléchir à une possible traduction littérale pour identifier des procédés de traduction suivis par le traducteur humain. Par ailleurs, une fonction pourrait être ajoutée à l'outil Yawat pour montrer où se situent les modifications à partir de la deuxième passe pour accélérer la révision à chaque nouvelle passe d'annotation.

Statistiques Nous présentons dans la figure 5 les statistiques portant sur les changements effectués pendant notre processus d'annotation en trois passes sur un sous-corpus [14]. Les chiffres dans la dernière colonne de la figure 5(b) signifient que, par exemple, le second annotateur était d'accord pour les 37 instances de type *Mod+Trans*, mais que le premier annotateur a corrigé ses premiers choix lors de la troisième passe. Ces deux tables montrent que les désaccords sur les frontières et sur les types diminuent progressivement grâce à cette procédure à plusieurs passes. Des exemples pour certains types de changements sont présentés dans la table 2.

À ce jour nous avons annoté un corpus de contrôle anglais-français, cinq sous-corpus anglais-français et deux sous-corpus anglais-chinois, dont les statistiques apparaissent dans la table 3. La table 4 donne elle les statistiques sur le nombre de tokens annotés par langue et par type pour la paire anglais-français [15]. Il apparaît que 73,5% des tokens anglais sont annotés avec les types *Littérale* et *Équivalence*, et 20,4% sont annotés avec *Modulation, Transposition, Mod+Trans, Généralisation* et *Particularisation*, qui sont les types de traduction les plus intéressants pour la génération de variantes non paraphrastiques par pivot.

Catégorie	Exemples
Littérale	I'll explain **it** -> je vais **l'**expliquer, refuses -> refuse de
Équivalence	here's -> voilà, no -> pas de
Particularisation	extend **it** -> allonge **la suite**
Modulation	we -> on, you -> on, about it -> en the reason [...] **is** -> la raison [...], **c' est** I encourage **all of you** -> je **vous** encourage **tous**
Transposition	humanity **'s** legacy -> héritage **de** l' humanité

TABLE 1: Exemples de conventions d'annotation.

Changement	Exemples
Étendre la frontière	the arctic ice cap **is, in a sense, -> on peut voir** la calotte glacière arctique **comme** (inclure "on peut voir")
Littérale -> Équivalence	global warming pollution -> pollution à effet de serre
Rajouter *Simplification*	**most of** the last three years -> ces 3 dernières années
Littérale -> Modulation	shallow -> peu profond
Modulation -> Transposition	increasing rapidly -> en augmentation rapide
Incertain -> Généralisation	sea change -> changement de tendance
Modulation -> Mod+Trans	the proposal has been to -> ils projettent de

TABLE 2: Exemples de changements de types survenus lors d'une deuxième passe.

	Nb lignes	Nb Tokens EN	Nb Tokens FR	Nb Caractères ZH
1	95	1,792	1,774	2,388
2	106	2,282	2,545	3,851
3	101	2,189	2,357	-
4	92	1,381	1,489	-
5	133	2,566	2,766	-
contrôle	100	3,055	3,238	-

TABLE 3: Statistiques sur les sous-corpus annotés.

	Anglais	Français	% EN tokens
Littérale	6,864	7,154	67,23%
Équivalence	645	822	6,32%
Modulation	1,173	1,221	11,49%
Transposition	189	263	1,85%
Mod+Trans	250	301	2,45%
Généralisation	172	121	1,68%
Particularisation	303	421	2,97%
Idiome	4	6	0,04%
Métaphore	16	19	0,16%
Simplification	122	0	1,20%
Explicitation	0	119	0,00%
Incertain	114	131	1,12%
Tous les types	9,852	10,578	96,49%
Aucun Type	358	353	3,50%
Nb tokens total	10,210	10,931	-

TABLE 4: Statistiques sur les annotations anglais-français (nombre de tokens).

14. Pour les 17 cas de désaccords de *Transposition* avec la même frontière, 11 cas concernent un changement de ponctuation, qui ont été par la suite annotés en *Modulation* selon une nouvelle convention rendue nécessaire.

15. Les chiffres ne sont pas encore définitifs, les deux derniers sous-corpus n'ayant pas encore subit une troisième passe d'annotation.

passe 1 à passe 2					
		même frontière		frontière différente	
	nb d'instances	même type	type différent	même type	type différent
Littérale	1443	1411	29	0	3
Équivalence	66	60	5	1	0
Généralisation	17	11	3	2	1
Particularisation	29	23	5	0	1
Modulation	54	50	4	0	0
Transposition	41	17	17	2	5
Mod+Trans	88	58	20	2	8
Idiome	2	1	1	0	0
Métaphore	5	0	5	0	0
Explicitation	3	3	0	0	0
Simplification	4	4	0	0	0
Incertain	17	13	4	0	0
Total	1769	1651	93	7	18

(a) Passe 1 à passe 2

passe 2 à passe 3								
	même frontière			frontière différente				
	total	accord	désaccord	total	AFAT	AFDT	DFDT	correction du premier annotateur
Littérale	**29**	22	7	**3**	2	1	0	3
Équivalence	**5**	3	2	**1**	0	1	0	3
Généralisation	**3**	2	1	**3**	1	1	1	0
Particularisation	**5**	4	1	**1**	1	0	0	0
Modulation	**4**	4	0	**0**	0	0	0	1
Transposition	**17**	17	0	**7**	6	0	1	3
Mod+Trans	**20**	16	4	**10**	8	2	0	37
Idiome	**1**	1	0	**0**	0	0	0	0
Métaphore	**5**	5	0	**0**	0	0	0	0
Incertain	**4**	4	0	**0**	0	0	0	0

(b) Passe 2 à passe 3

FIGURE 5: Statistiques sur les changements de types (nombre d'instances) pendant un processus d'annotation en trois passes sur un sous-corpus. AFAT : accord sur la frontière et le type ; AFDT : accord sur la frontière mais avec type différent ; DFDT : différente frontière et différent type.

Analyse contrastive entre langues cible L'annotation de ce corpus multilingue parallèle permet de révéler des contrastes entre les traductions vers des langues différentes (voir figure 6). Nous présentons des statistiques préliminaires basées sur l'annotation de deux sous-corpus anglais-français et anglais-chinois (voir table 3), et nous comparons la traduction des segments anglais strictement identiques (i.e. avec la même frontière). Nous trouvons ainsi qu'il existe moins de traductions littérales vers le chinois, et que la différence principale avec la traduction vers le français consiste en l'utilisation de différents types de *Modulation*. Les traducteurs chinois ont beaucoup plus recours à des phénomènes d'*Explicitation*. Nous poursuivons l'annotation de la paire anglais-chinois pour établir une analyse plus fine des contrastes obtenus.

	Anglais	Français	Anglais	Chinois
Littérale	2796	2896	2141	3414
Équivalence	247	310	326	478
Modulation	343	357	386	549
Transposition	106	150	112	171
Mod+Trans	128	126	29	47
Généralisation	58	37	158	154
Particularisation	132	180	210	505
Idiome	0	0	0	0
Métaphore	10	15	6	10
Simplification	74	-	178	-
Explicitation	-	46	-	459
Incertain	35	31	128	238
Tous les types	3929	4148	3674	6025
Aucun Type	145	171	400	214
Nb tokens total	4074	4319	4074	6239

FIGURE 6: Table de contraste entre les traductions vers le français et vers le chinois (nombre de tokens).

Nous constatons que parfois la qualité de la traduction chinoise n'est pas aussi bonne que la traduction française dans notre corpus, pour des raisons multiples : manque de connaissances du domaine spécifique d'une intervention ; manque d'édition finale pour corriger des erreurs évidentes, résultant notamment en des mots anglais laissés non traduits, des traductions erronées (voir figure 6) ou des ponctuations absentes. En outre, des traductions extrêmement libres posent même de réelles difficultés pour l'alignement manuel de mots.

Un dernier point qui n'est pas illustré dans la figure 6 concerne l'introduction d'idiomes en chinois pour traduire des expressions non figées. Par exemple :

bring our children into the world -> 生儿育女 (give birth to and raise children)

forest ***upon which the people depend*** *->* 栖身之所 *(shelter)*
pressured the people a little bit about it -> 刨根问底 *(inquire into the root of the matter)*

L'utilisation d'idiomes chinois en traduction est considérée comme une bonne pratique qui permet d'obtenir des textes concis adaptés à la culture chinoise. Puisque ces idiomes peuvent être traduits de différentes manières plus ou moins libres, ils peuvent contribuer de façon importante à l'obtention de paraphrases par pivot.

Le corpus complet (2 436 lignes dans chaque langue) sera distribué dans un format XML avec les métadonnées d'accord inter-annotateurs et de modifications entre passes d'annotation.

6 Conclusion et perspective

Dans ce travail nous nous sommes intéressée aux relations de traduction car celles-ci n'avaient pas jusque-là été prises en compte à notre connaissance dans l'approche de génération de paraphrases par équivalence de traduction, ou pivot, ainsi qu'en traduction automatique. Nous avons catégorisé ces relations et les avons annotées dans un corpus parallèle multilingue de *TED Talks*. Nous avons choisi ce genre spécifique (discours transcrit et traduit) afin d'obtenir plus de diversité qu'avec des corpus techniques. Le travail d'annotation est en phase préliminaire pour finaliser le guide d'annotation. L'accord inter-annotateurs mesuré est fort pour les segments de mêmes frontières, mais nous avons adopté un processus d'annotation à plusieurs passes, plus coûteux en temps, afin de garantir une meilleure qualité d'annotation. La faisabilité de la tâche étant confirmée, nous étendrons les annotations sur le corpus entier afin de les mettre à disposition de la communauté.

À court terme, nous allons réaliser des annotations plus fines sur les blocs de type *Modulation, Transposition* et *Modulation+Transposition*. Pendant les premières annotations, nous avons en effet privilégié l'objectif de garder une information complète à celui d'obtenir un alignement des unités les plus petites possibles, comme illustré par les paires de segments suivantes : *is believed in enough -> peut être tellement crédible*, *make it seem -> lui donner l'apparence*, *they're able to be moved around -> on peut les déplacer*, *have this kind of reception -> être reçu de cette manière*, *give you good close look at this -> vous montrer de près*. Cependant, ces segments particuliers sont souvent peu réutilisables, et il est nécessaire de réaliser un alignement plus fin pour détailler leur structure, notamment dans le but d'apprendre des patrons puis de vérifier si ceux-ci ont des attestations dans des corpus parallèles de grande taille.

Une fois ce corpus annoté finement avec l'ensemble des relations de traduction, nous développerons un classifieur pour les détecter automatiquement. De telles informations n'ont pas été prises en compte à notre connaissance dans les travaux précédents portant sur la génération de paraphrases ou la traduction automatique.

Références

APIDIANAKI M., VERZENI E. & MCCARTHY D. (2014). Semantic clustering of pivot paraphrases. In N. CALZOLARI, K. CHOUKRI, T. DECLERCK, H. LOFTSSON, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Eds., *Proceedings of the Ninth International Conference*

on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014., p. 4270–4275 : European Language Resources Association (ELRA).

BANNARD C. J. & CALLISON-BURCH C. (2005). Paraphrasing with bilingual parallel corpora. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, p. 597–604.

BARZILAY R. & MCKEOWN K. (2001). Extracting paraphrases from a parallel corpus. In *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference, July 9-11, 2001, Toulouse, France.*, p. 50–57.

CALLISON-BURCH C. (2008). Syntactic constraints on paraphrases extracted from parallel corpora. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, p. 196–205.

CETTOLO M., GIRARDI C. & FEDERICO M. (2012). Wit^3 : Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, p. 261–268, Trento, Italy.

CHUQUET H. & PAILLARD M. (1989). *Approche linguistique des problèmes de traduction anglais-français*. Ophrys.

COCOS A. & CALLISON-BURCH C. (2016). Clustering paraphrases by word sense. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL 2016)*, San Diego, California : Association for Computational Linguistics.

COHEN J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**, 37–46.

DENG D. & XUE N. (2017). Translation divergences in chinese-english machine translation : An empirical investigation. *Computational Linguistics*, **43**(3), 521–565.

DOLAN B., QUIRK C. & BROCKETT C. (2004). Unsupervised construction of large paraphrase corpora : Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA : Association for Computational Linguistics.

DYER C., CHAHUNEAU V. & SMITH N. A. (2013). A simple, fast, and effective reparameterization of IBM model 2. In L. VANDERWENDE, H. D. III & K. KIRCHHOFF, Eds., *Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, p. 644–648 : The Association for Computational Linguistics.

GANITKEVITCH J. & CALLISON-BURCH C. (2014). The multilingual paraphrase database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, p. 4276–4283.

GANITKEVITCH J., DURME B. V. & CALLISON-BURCH C. (2012). Monolingual distributional similarity for text-to-text generation. In E. AGIRRE, J. BOS & M. T. DIAB, Eds., *Proceedings of the First Joint Conference on Lexical and Computational Semantics, *SEM 2012, June 7-8, 2012, Montréal, Canada.*, p. 256–264 : Association for Computational Linguistics.

GANITKEVITCH J., DURME B. V. & CALLISON-BURCH C. (2013). PPDB : the paraphrase database. In *Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, p. 758–764.

GERMANN U. (2008). Yawat : Yet another word alignment tool. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Demo Papers*, p. 20–23 : The Association for Computer Linguistics.

IBRAHIM A., KATZ B. & LIN J. (2003). Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second International Workshop on Paraphrasing - Volume 16*, PARAPHRASE '03, p. 57–64, Stroudsburg, PA, USA : Association for Computational Linguistics.

KOK S. & BROCKETT C. (2010). Hitting the right paraphrases in good time. In *Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, p. 145–153.

LAPATA M., SENNRICH R. & MALLINSON J. (2017). Paraphrasing revisited with neural machine translation. In M. LAPATA, P. BLUNSOM & A. KOLLER, Eds., *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1 : Long Papers*, p. 881–893 : Association for Computational Linguistics.

LI Z. & SUN M. (2009). Punctuation as implicit annotations for Chinese word segmentation. *Comput. Linguist.*, **35**(4), 505–512.

LIN D. & PANTEL P. (2001). DIRT – discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, p. 323–328, New York, NY, USA : ACM.

MILLER G. A. (1995). Wordnet : A lexical database for english. *Commun. ACM*, **38**(11), 39–41.

MONTI J., SANGATI F. & ARCAN M. (2015). Ted-MWE : a bilingual parallel corpus with MWE annotation towards a methodology for annotating mwes in parallel multilingual corpora.

PANG B., KNIGHT K. & MARCU D. (2003). Syntax-based alignment of multiple translations : Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, p. 102–109, Stroudsburg, PA, USA : Association for Computational Linguistics.

PAVLICK E., BOS J., NISSIM M., BELLER C., DURME B. V. & CALLISON-BURCH C. (2015a). Adding semantics to data-driven paraphrasing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1 : Long Papers*, p. 1512–1522.

PAVLICK E., RASTOGI P., GANITKEVITCH J., DURME B. V. & CALLISON-BURCH C. (2015b). PPDB 2.0 : Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2 : Short Papers*, p. 425–430.

WU Y., SCHUSTER M., CHEN Z., LE Q. V., NOROUZI M., MACHEREY W., KRIKUN M., CAO Y., GAO Q., MACHEREY K., KLINGNER J., SHAH A., JOHNSON M., LIU X., KAISER L., GOUWS S., KATO Y., KUDO T., KAZAWA H., STEVENS K., KURIAN G., PATIL N., WANG W., YOUNG C., SMITH J., RIESA J., RUDNICK A., VINYALS O., CORRADO G., HUGHES M. & DEAN J. (2016). Google's neural machine translation system : Bridging the gap between human and machine translation. *CoRR*, **abs/1609.08144**.

ZHAO S., WANG H., LIU T. & LI S. (2008). Pivot approach for extracting paraphrase patterns from bilingual corpora. In K. MCKEOWN, J. D. MOORE, S. TEUFEL, J. ALLAN & S. FURUI, Eds., *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, p. 780–788 : The Association for Computer Linguistics.

Annotation automatique d'images: le cas de la déforestation

Duy Huynh[1] Nathalie Neptune[1]
(1) Institut de Recherche en Informatique de Toulouse, UMR5505 CNRS
Université de Toulouse
118 Route de Narbonne, F-31062 Toulouse Cedex 9, France
duy.huynh@irit.fr, nathalie.neptune@irit.fr
Under the supervision of
Josiane MOTHE (1)

RÉSUMÉ
Cet article correspond à un état de l'art sur le thème de l'annotation automatique d'images d'observation de la terre pour la détection de la déforestation. Nous nous intéressons aux différents challenges que recouvre le domaine et nous présentons les méthodes de l'état de l'art puis les pistes de recherche que nous envisageons.

ABSTRACT
Automatic image annotation : the case of deforestation.

This paper aims to present the state of the art of the methods that are used for automatic annotation of earth observation image for deforestation detection. We are interested in the various challenges that the field covers and we present the state of the art methods and the future research that we are considering.

MOTS-CLÉS : Recherche d'information, annotation d'images, réseaux de neurones convolutionnels, détection de la déforestation.

KEYWORDS: Information retrieval, image annotation, CNN, convolutional neural networks, deforestation detection.

1 Introduction

According to the National Geographic, forest covers about 30% of the planet. Forest ecosystems play an essential role in supporting life on the earth such as supplying wood, water regulation, preventing storms and soil erosion and forests store rare genetic resources for our planet. Deforestation affects the environment in a multitude of ways. The most obvious effects are global warming and loss of biodiversity. From the photosynthetic function of trees, forests release oxygen and absorb carbon dioxide. The fewer forests, the more carbon dioxide entering the atmosphere, increasing the speed of global warming. In addition, earth's forests are home to over 80% of plants and animals but deforestation destroys these habitats, diminishing biodiversity and causing the extinction of four to six thousand rainforest species every year (Geographic, 2017). Direct causes of deforestation are agricultural expansion, logging and wood extraction, bio fuels from palm oil, infrastructural expansion such as road and urbanization, and mining (Tariq & Aziz, 2015) (see also http://www.

theworldcounts.com/stories/deforestation-facts-and-statistics).

Satellite remote sensing makes it possible to observe the earth and help detect deforestation in a faster way than ever. For example, the Copernicus program provides images from any region every 5 days [1]. Moreover, with the advances of computing power and machine learning techniques, it becomes possible to detect deforestation automatically from earth observation (EO) images (Achard *et al.*, 2002; O'Connor *et al.*, 2015).

This paper reviews the state of the art of the methods that are used in the domain of automatic deforestation detection. When using remote sensing, deforestation detection is an application of change detection. It generally consists in two steps : the annotation of satellite images in order to identify the land cover and the change detection on zones that are identified as forest. In these two steps, convolution neural networks (CNN) were proven successful. This is because CNN allow for a large learning capacity(Krizhevsky *et al.*, 2012) ; moreover their automatic feature extraction capability is very useful in image analysis (Witten *et al.*, 2016). Figure 1 shows the basic workflow of automatic deforestation detection using two images of the same area captured at different times. The first step is linked to information retrieval and can be paralleled with the tasks developed in ImageCLEF for automatic association of concepts to images in medicine (Ionescu *et al.*, 2017), although it goes a step further since image segmentation is usually considered to annotate the images in the case of deforestation detection while it is usually an index only in information retrieval.

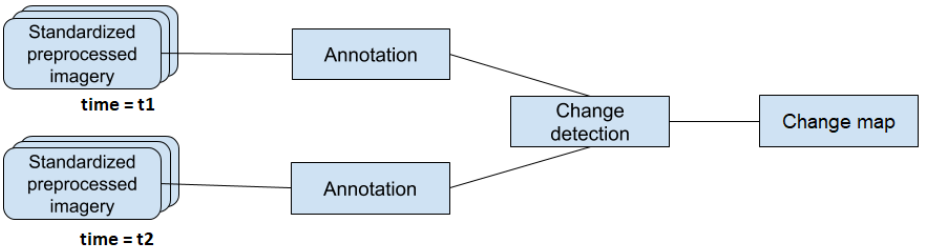


FIGURE 1: Basic workflow for annotation and change detection in satellite imagery of the same area taken at two different times.

The remainder of this paper is as follows : sections 2 and 3 present the state of the art of the two steps of the automatic deforestation detection problem. In section 4, we present the main current challenges which also correspond to the ones that we would like to tackle in our future work. Section 5 concludes this paper. Additionally, a brief explanation of how convolutional neural networks work is presented in the appendix.

1. "Copernicus, previously known as GMES (Global Monitoring for Environment and Security), is the European Programme for the establishment of a European capacity for earth Observation." www.copernicus.eu

2 Annotation of land cover

Image annotation is a general problem which is not specific to forest detection but rather that covers a large variety of domains such as information extraction from medical images for disease detection (Ionescu *et al.*, 2017),(Mothe *et al.*, 2017), and image retrieval (Babenko *et al.*, 2014). In the case of deforestation detection, what we need is land cover information. The automatic annotation of earth observation images to associate land cover types with areas is typically done with classification methods. Our intention is not to provide a state of the art of image classification methods but rather to detail how these methods have been used in the case of land cover annotation with an emphasis on deforestation detection.

As in many fields, deep learning has revolutionised the domain of image annotation for deforestation detection in remote sensing. Without aiming to be exhaustive, in this section, we first present some of the approaches that have been developed for land cover annotation of satellite imagery before the deep neural networks arose, then we present how deep learning is used in this domain.

2.1 Methods used before deep learning

Classification-based methods are very commonly used when having to annotate satellite images for deforestation detection. (Shimabukuro *et al.*, 1998) and (Müller *et al.*, 2016) are two examples of such methods, the first one opting for an unsupervised classification approach by segment while the second one is using a supervised pixel-by-pixel approach.

To generate deforestation maps and provide related information on areas experiencing deforestation, (Shimabukuro *et al.*, 1998) proposed an approach based on shade fraction image generated by a spectral mixture model, then "region growing" segmentation and unsupervised classification (clustering) of fields were applied. At the time, the state of the art was visual interpretation of satellite imagery or classification based on pixel-by-pixel analysis without contextual information. However, visual interpretation is laborious and thus costly especially when dealing with a large number of small surfaces to label in the same area. Using shade fraction images instead of working with the Landsat images directly allowed for reduced processing time for classification and post-processing time for manual removal of undetermined classes. Their results were validated against results from conventional techniques in use at the time on images from Rondônia, in Brazil.

Yearly deforestation patterns for an area of the Amazon forest in Mato Grosso and Pará were derived from Landsat TM [2] and ETM+ [3] images captured between 1985 and 2012, by (Müller *et al.*, 2016). The proposed approach uses a random forest classifier on the training data made of labelled samples of the values of reflectance images and minimum tasseled cap wetness (TCW) observations, for each pixel. An overall accuracy of 85% was reached with 95% confidence interval margin of $\pm 2\%$. Classification error tended slightly towards late detections. Higher deforestation rates were found compared to the state of the art deforestation datasets for the same region and over the same time

2. Landsat Thematic Mapper (TM) is a an advanced multi-spectral scanning sensor carried by Landsat 4 and 5 and featuring seven spectral bands. With band 6 being a thermal infrared radiation sensor. The TM sensor has a spatial resolution of 30 m X 30 m (120 m for band 6) and a temporal resolution of 16 days. https://landsat.gsfc.nasa.gov/the-thematic-mapper/

3. Landsat Enhanced Thematic Mapper Plus (ETM+) is an enhanced version of Landsat TM carried by Landsat 7. ETM+ is a multi-spectral scanning radiometer with eight bands. It has a spatial resolution of 30 m X 30 m (60 m for the thermal band, 15 m for the panchromatic band). https://landsat.gsfc.nasa.gov/the-enhanced-thematic-mapper-plus/

period.

The next subsections review the approaches in the literature that use convolutional neural networks (CNN) to annotate remote sensor data, in particular for the detection and analysis of deforestation.

2.2 Methods using convolutional neural networks

Several approaches using neural networks, including convolutional neural networks have been proposed for annotating satellite images, such as (Kussul *et al.*, 2017) and (Zhang *et al.*, 2017). In the first approach the annotation is done by segment while it is done by pixel in the second approach.

(Kussul *et al.*, 2017) proposed a deep learning approach for the classification of multi-source satellite images. They used a combination of supervised and unsupervised neural networks to segment and classify satellite imagery from Landsat 8 and Sentinel 1A. Testing was done with data from the Joint Experiment of Crop Assessment and Monitoring (JECAM) test site in Ukraine. While this study focused mostly on crop identification, the area was labelled with eleven land cover types among which the "forest" type. This approach was tested for overall accuracy against the following approaches : random forest (RF), and an ensemble of neural networks (ENN) made of multilayer perceptrons. The proposed ensemble of 2D CNN outperformed these two other methods in overall accuracy. However, for the forest class, the difference is only a few decimal points because the random forest and ENN had already reached over 99% accuracy.

The approach from (Zhang *et al.*, 2017) classifies fine resolution images from remote sensors with a model integrating a CNN and a multilayer perceptron (MLP). This model was compared to standalone standard pixel-based CNN and MLP classifiers and a pixel-based texture MLP based on the standard Grey Level Co-occurrence Matrix (GLCM-MLP). The authors tested the method on data from Southampton and its surroundings, in the UK. A total of eight land cover classes were detected and large patches of forest in rural areas. While they did not have a forest class, forest patches were put in the "Trees" class which is described as "large patches of deciduous trees" and "patches of tree species". The overall accuracy of the MLP-CNN approach, including for the Trees class, on the two test sites was found to be higher than that of the other three methods.

In the next subsection we discuss a Kaggle [4] competition aimed at automatically classifying forest images for deforestation detection and the prevalence of CNN models.

2.3 The Kaggle competition for deforestation detection

Deforestation can also be detected in a single image by detecting known deforestation patterns in a forested area. The "Planet Deforestation Detection" Kaggle competition launched by Planet Labs [5] in April 2017, proposed such task.

The competition aimed to find effective methods to track forest changes by using high resolution satellite imagery. The images were provided as image chips and the task was to automatically annotate each image chip with its corresponding labels for atmospheric condition and land cover. A total of 17 possible labels were defined. The competition started on April 20, 2017 and ended on July 20, 2017 (Kaggle, 2017).

4. Kaggle is a data science and machine learning competition platform. `https://www.kaggle.com/`
5. Planet Labs is an earth observation satellite company based in San Francisco `https://www.planet.com/`

The chips are extracted from satellite images and represent approximately one square kilometre (Scott, 2017). These chips are from Planet's full-frame analytic scene products and were provided in GeoTiff formats with four bands of data and in JPG format. The scenes included were exclusively from the Amazon basin (**?**). The training set contained over 40000 images while the test set contained over 20000 images.

The F2-score, which is a weighted average of precision and recall, was used to rank the submissions.

The formula for calculating the F2-score was given as follows (Kaggle, 2017) :

$$(1+\beta^2)\frac{P.R}{\beta^2 P+R} \text{ where } P=\frac{TP}{TP+FP}, R=\frac{TP}{TP+FN}, \beta=2.$$

P is the precision and R is the recall, TP stands for true positives, FP for false positives and FN for false negatives, β is a float which in this case is equal to 2 to indicate that a higher weight is given to the recall than to the precision.

There were some data quality issues with TIFF images not matching their corresponding JPG images in the test set and there were labels incorrectly assigned as well. Ambiguous labelling was in several cases due to the fact that from the image alone certain classes could not be told apart by visual interpretation. The winning model used exclusively the JPG images with several data augmentation techniques to pre-process the images such as image rotation and haze removal (He *et al.*, 2011). A model made of 11 CNN was used with existing CNN architectures such as Inception, Resnet and Densenet. Then ridge regression was used on the probabilities obtained for each label. The CNN were then combined with a ridge regression model. Finally, the loss function was designed to take into account the F2-score used for evaluating the submissions [6] (Kaggle, 2017).

A total of 938 submissions were made for this competition (Kaggle, 2017). Of the top 16 teams from the private leader-board, 7 reported using CNN models [7]. (Lagrange *et al.*, 2015) had found that for semantic labelling of earth observation images, the best performance is obtained when using deep convolutional neural networks compared to expert classifiers and spectral support-vector classification.

Image annotation is the first step in change detection, and deep learning can be used for automatically learning features and annotating the images. The result of the image annotation task is images that are semantically labelled and can be compared to detect changes (Hirschmugl *et al.*, 2017). In the next section we will go over the methods that are commonly used for change detection in remote sensing imagery.

3 Change detection in remote sensing images

In remote sensing, the use of satellite images to assess and then map deforestation is one of the applications of change detection. Two methods are applied for change detection with optical images : image to image change detection and time series analysis. Image-to-image change detection requires a minimum of two images captured at different times while time series analysis requires a series

6. http://blog.kaggle.com/2017/10/17/planet-understanding-the-amazon-from-space-1st-
7. https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/discussion/36732

of images captured over a period of time. Image-to-Image change detection is more commonly used. Performing change detection over long time periods may require the use of images captured by different remote sensing devices with different characteristics which then requires data fusion techniques to combine these data together.

3.1 Methods prior to deep learning

Deforestation detection is one of the major applications of satellite image change detection. The problem has been studied for decades and many techniques have been proposed. These techniques can be categorized into different approaches such as algebra, transformation, classification, advanced models, geographical information system (GIS) approaches, or visual analysis (Lu *et al.*, 2004).

The very first *algebra technique* was univariate image differencing. This straightforward technique detects the change by applying a threshold to the difference in pixel value between first-date image and the second-date image (Lu *et al.*, 2004). This technique was widely used in change detection problems, particularly for detecting forest changes as in (Miller *et al.*, 1978) and others (Singh, 1989). Another well-known algebra technique was image regression. First, the method assumed that pixels in the same location are related by a linear function in time. Thus, the pixels values in the second-date image can be predicted according to the regression function. Finally, a threshold was applied to the difference between the true second-date value and the predicted second-date value. This technique showed better performance than the image differencing technique (Singh, 1989).

The second group of techniques uses *transformations* such as Principal Component Analysis (PCA), Multitemporal Kauth-Thomas (MKT), Gramm–Schmidt (GS), and Chi-square transformations (Lu *et al.*, 2004).

(Collins & Woodcock, 1996) examined PCA, MKT, GS methods to the problem of forest change due to conifer mortality and concluded that PCA and MKT give better results than GS.

The third group of change detection techniques is made of classification approaches which have been used for both image to image change detection and for time series change detection like in (Mertens & Lambin, 1997) and (Olofsson *et al.*, 2016) respectively.

(Mertens & Lambin, 1997) proposed a model to detect deforestation in southern Cameroon based on remote sensing data from Landsat MSS sensor [8] for the years 1973 and 1986. Photographs, digitized topographic maps, aerial photos and other remote sensing data and population data were used as well. In addition, ground observation data were collected and used to validate the land cover and land cover change maps. A maximum likelihood classifier was used to generate land cover maps for 1973 and 1986, independently for the areas common to the two Landsat images. The classifier was trained on field observations and interpretations of aerial photos of unchanged areas for 1979. The accuracy was evaluated with field observation data and low-altitude aerial photos. For the classification task, 90% accuracy was reached for 1973 but only 85% for 1986 due to haze. Five classes were derived among which the "dense forest" which the authors define as "evergreen or moist deciduous forest zones, dominated by trees at least 5 m high and with a forest-cover proportion of 30 per cent or more", the four other classes were for different non-forested areas.

8. The Landsat MSS sensor is a multi spectral scanner carried by Landsat satellites one through five, with a spatial resolution of 68 m X 83 m and a temporal resolution of 18 days. https://landsat.gsfc.nasa.gov/the-multispectral-scanner-system/

A time series analysis was performed by (Olofsson *et al.*, 2016) to reveal deforestations trends in the New England area in the North East of the United States with the goal of modelling the impact that forest changes have on the carbon balance of the planet. To map the land cover change, the Continuous Change Detection and Classification (CCDC) algorithm was used on pixel-level time series of Landsat data available for the area from 1985 to 2011. Areas of forest harvest were not added to the deforestation estimate. A random forest classifier was used on the training data with the attributes from the time series prediction model for each time series. To account for bias caused by classification errors, the mapped area is estimated from a random sample of reference observations then visual examination is performed to confirm the labels. This study detected the land cover changes but also capture the evolution of forest loss over time. Accelerated deforestation was found to have taken place in the 1990 then around 2007 the deforestation rate stabilized.

While change detection with optical imagery is fairly common, less work has been done with radar imagery especially with time series. However there have been some contributions on the subject such as in (Barreto *et al.*, 2016).

A change detection method based on object detection for high resolution Synthetic Aperture Radar (SAR) data was proposed by (Barreto *et al.*, 2016). This method involves three steps. First, multi-temporal Xband high resolution SAR image segmentation, followed by feature extraction and finally, area detection and classification. The image is segmented into superpixels with a simple linear iterative clustering algorithm (SLIC). The features are extracted with the object correlation images (OCI) framework and with gray-level cooccurrence matrix (GLCM). Areas are detected and classified into unchanged, deforestation and other changes classes with a multilayer perceptron. Experts manually annotated a set of multi-temporal Xband SAR images captured from August to November 2015 with 2,263 regions of interest. Results showed an improvement of 10% in accuracy, compared to the state of the art approaches, in change detection and classification for the deforested areas. The proposed approach was compared to the following state of the art object-based approaches : Optimum Path Forest (OPF) clustering for image segmentation, OCI alone for feature extraction, SVM and OPF-classifier for classification.

3.2 Method using deep learning

(Khan *et al.*, 2017) proposed a new deep CNN model for object-based change detection in incomplete satellite imagery. Their approach performs two tasks, first data recovery to fill data that is missing due to limited camera aperture, cloud cover, and sensor artefacts, then change detection which is treated as a regions classification problem. For this second task, object-based change detection was used without domain knowledge to extract features, instead, all features were learned with a deep neural network.

The model was used to analyse satellite data on the north-east region of Melbourne, in Victoria, Australia. The authors were able to detect forest cover change on images taken from 1987 to 2015. This approach outperforms baseline techniques in temporal change detection and patch-wise classification tasks.

For start-time and end-time predictions for detected change events, this approach outperforms the state-of-the-art approaches, one based on hand-crafted features for classification and the second one based on bag of visual words for classification. To establish the baseline with these two state-of-the-art approaches, dense scale invariant feature transform (SIFT) descriptors were used as a baseline for

change detection and linear SVM, kernel SVM and random forest (RF) were used for prediction.

3.3 Summary of state of the art approaches and cited papers

Table 1 shows the papers that were cited for annotation and change detection with earth observation images. The papers are categorized by the type of approach used. Among the papers cited, most have used a classification-based approach, including the ones using CNN.

TABLE 1: Publications cited and the methods that they propose for each annotation and change detection tasks.

METHOD	TASK	
	Annotation	**Change detection**
Algebra based approach		(Miller *et al.*, 1978)
Classification based	*Prior to CNN*	
	(Miller *et al.*, 1978) (Mertens & Lambin, 1997) (Shimabukuro *et al.*, 1998) (Müller *et al.*, 2016) (Olofsson *et al.*, 2016) (Barreto *et al.*, 2016)	(Müller *et al.*, 2016) (Olofsson *et al.*, 2016) (Barreto *et al.*, 2016)
	CNN	
	(Khan *et al.*, 2017) (Kussul *et al.*, 2017) (Zhang *et al.*, 2017) (Kaggle, 2017)	(Khan *et al.*, 2017)
Transformation		(Collins & Woodcock, 1996) (Müller *et al.*, 2016)
Advanced model (spectral mixture)	(Shimabukuro *et al.*, 1998)	
Other methods	(Achard *et al.*, 2002)	(Achard *et al.*, 2002)

Deforestation detection using earth observation data implies mainly two complementary tasks : (a) detecting in a given image what is forest and what is not (i.e. image annotation) (b) detecting changes between images. While deep learning model have shown promising results in these two tasks, remote sensing suffers from a limited availability of annotated data for training such models. Moreover, it is likely that using different sources of evidence can help in the accuracy of the detection. The next section is related to these issues.

4 Research questions

From our readings and from related work, we elaborate what we think are the main research questions at this stage in the domain of automatic detection of deforestation using earth observation imagery.

4.1 Data fusion : how to combine various sources of evidence ?

A current trend in remote sensing is data fusion (Zhang, 2010). The main idea is to combine data from multiple sources in order to produce better quality results taking advantage of the information each source carries. While data fusion is not new in remote sensing (Pohl & Van Genderen, 1998), the most recent research in the field focuses on high-level fusion in place of pixel-level fusion. (Joshi *et al.*, 2016) reviewed 112 studies for various types of applications on fusing optical and radar data and concluded that the main methods used are pre-classification fusion followed by pixel-level inputs in traditional classification algorithms.

Various types of data can be fused. For example, (Reiche *et al.*, 2015) presented a fusion approach (MulTiFuse) that exploits the full observation density of optical and Synthetic Aperture Radar (SAR) time series to detect deforestation in the tropics. (Schmidt *et al.*, 2015) used coarse spatial resolution MODIS data combined with finer spatial resolution Landsat data to map forest and agricultural elements of an area in central southeast Queensland, Australia. In (Reiche *et al.*, 2018), the authors show that spatial and temporal accuracies of the multi-sensor approach were higher than the single-sensor results for near real-time deforestation detection in tropical dry forests (the authors combined Sentinel-1 C-band SAR time series with ALSO-2 PALSAR-2 L-band SAR, and Landsat-7/ETM+ and 8/OLI).

Current data fusion techniques focus on fusing data at various levels of resolution, various number of layers for multi-spectral imagery but a few consider other sources of evidence such as statistics, scientific publications, etc. While satellite data is one source of evidence for deforestation detection, it may not be enough to explain the cause of deforestation and monitor its evolution as well as other related geo-phenomena. To get a step further we would like to combine earth observation imagery with other sources of evidence such as statistics related to wood exportation, type of manufacturing goods implying wood or derivatives, level of urbanisation and population growth. Another source of evidence we would like to use in cross analyses are reports and scientific papers. We believe that scientific papers on sustainable development contain relevant information about deforestation and the related geo-phenomena and that this information can be extracted and combined with data from earth observation imagery for analysis purposes. The main issue to fuse these sources is their high level of heterogeneity which is an open question.

4.2 Transfer learning with CNN : how effective is it ?

Transfer learning is the process of learning features by training a network on a large data set and then transferring these features to a different dataset. This approach has also been used in remote sensing and showed promising results by (Hu *et al.*, 2015) and (Salberg, 2015). The ability for learned features to be transferred across domains accounts for the success of transfer learning in deep learning and makes it a promising approach to cope with the problem of limited training data. Transfer learning is also used to overcome the problem of the lack of annotated data for training.

The annotation of remote sensing images is done either by human annotators or automatically by a computer program. In both cases, ground truths are the reference against which these annotations can be validated. However, going on the ground to collected theses ground truths may be costly and impractical. Consequently, the amount of labelled data with validated labels remains limited compared to the large amount of earth observation data that is available.

To overcome the problem of limited labelled data available for training, several approaches using transfer learning have been proposed in the remote sensing field. The effectiveness of deep learning approaches in remote sensing considering the insufficient training data is still an open question. So far, limited work has been done on change detection in forests with deep learning and CNN in particular. We aim to propose a new approach to detect and map deforestation in tropical forests using transfer learning.

Our goal is to propose a model to track the evolution of deforestation over time with time series analysis. By using transfer learning, we aim to find the best suited image dataset for this task by testing with various types of images.

We aim to provide a general model that can be reused in different countries and areas. This will require training our model on a large and diverse dataset. There is a massive amount of spatial data available though mostly unlabelled. We therefore aim to propose an unsupervised approach to annotate these images and validate the results with very high resolution images and ground truth data when available.

4.3 How to adapt CNN architectures to multi-spectral images ?

Satellite images are multi-spectral and pose the challenge of high dimensionality for machine learning. We will propose a CNN model that is adapted to this particular type of data. We aim to propose a dimension reduction approach that is optimal for our problem of detecting and tracking deforestation in tropical forests.

4.4 How to get annotated satellite images for our areas of interest ?

Due to the fact that annotated satellite data for most tropical forests is scarce, another contribution we aim to make is provide a new data set of labelled earth observation data for one of our areas of interest. To achieve this, we will use very high resolution images and ground truth data from local organizations if available or from field missions.

5 Conclusion

This paper presented the state of the art related to automatic detection of deforestation using earth observation images from remote sensors. We present the methods that have been developed in the two main steps of automatic detection of deforestation : image annotation which includes image segmentation and classification, and change detection. For both steps, we chose to present first the methods that were developed before deep neural networks then the methods which use deep learning. As in many other fields, neural networks are now very commonly used in remote sensing. Finally, we mentioned the research questions we would like to tackle in our future work and for two of them, namely data fusion and transfer learning, we detailed related work.

Acknowledgements

The work of Nathalie Neptune is supported by the Schlumberger Foundation through the Faculty for the Future fellowship. We would also like to thank the reviewers for their valuable comments. This work also benefits support from FabSpace 2.0 project ; FabSpace 2.0 received funding from the European Union's Horizon 2020 Research and Innovation programme under the Grant Agreement number 693210.

6 Appendix - Convolutional neural networks

Convolutional Neural Networks are one of the deep neural network classes that give the best current results in most computer vision problems such as classification and object recognition (Krizhevsky *et al.*, 2012), (Witten *et al.*, 2016). Basically, CNN is a straightforward Artificial Neural Network (ANN) in which the architecture consists of several layers connected together in a multi-tiered structure (LeCun *et al.*, 1998). There are three main types of layers : convolution layer,activation layer, pooling layer and fully connected layer (Krizhevsky *et al.*, 2012).

Before going into the basic components of CNN, we would like to look at an example describing how a feedforward ANN processes an input information. From that, we can draw on the effects of CNN components.

Suppose that we have a 200 x 200 image processed by a fully connected ANN. Each neuron needs 40,000 parameters to be trained which is costly. To reduce the number of parameters it is necessary to reduce the number of connections between layers ; this is the objective of the CNN convolution component. The idea is that each neural only needs to be connected to a local area of the image instead of the entire image. This feature enlarges CNN learning capacity (Krizhevsky *et al.*, 2012) which is the basic need to deal with large dataset problem such as earth observation images.

According to the literature of the domain (see for example (Krizhevsky *et al.*, 2012) there are 4 types of layers in a CNN which are defined as follows :

Convolution layer
The convolution layer plays the main role in the architecture of a CNN. Each neuron of the layer is formed by doing a convolution between a kernel and an image, applying convolution to the images aims at extracting important features such as edges, direction, color (Witten *et al.*, 2016).

Activation layer
This layer is usually placed right after the convolutional layer. This layer uses an activation function such as sigmoid, tanh, softplus, rectified linear unit (ReLU). However, ReLU ($f(x) = \max(0, x)$) is used most recently. The function converts all negative values in the result obtained from the convolutional layer to the value 0. The meaning of this setting is to make the model non-linear (Witten *et al.*, 2016). Using an activation layer also increases the learning rate (Krizhevsky *et al.*, 2012).

Pooling Layer
The goal of this layer is to reduce the matrix size but still highlight the features that are present in the input matrix. Max pooling is often used (Guo *et al.*, 2016). In terms of meaning, Max-Pooling determines where the strongest signal is when applying a filter. It is done by taking the maximum value of the neurons within the pooling region (Witten *et al.*, 2016).

Fully Connected Layer
This layer is similar to the feedforward ANN : all the nodes of the current layer are fully linked to the nodes from the next layer. After the image is processed by the previous layers, the image data will no longer be too large compared to the ANN model. This layer is placed at the end part of the CNN (Witten *et al.*, 2016).

A convolution neural network is formed by putting the above layers together. The model always starts with the convolutional layer. The activation layer usually follows right after the convolutional layer or even merges both layers into a layer. The next layer can be convolutional or pooling. This pattern can be repeated depending on the architecture. The output of these layers then may feeded to fully connected layers. The final layer of CNN usually uses the softmax function ($\sigma\left(z\right)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$) which forces the output of the network represent a probability distribution across discrete alternatives (Krizhevsky *et al.*, 2012). Figure 2 represents a general CNN which uses the ReLU function as an activation function.

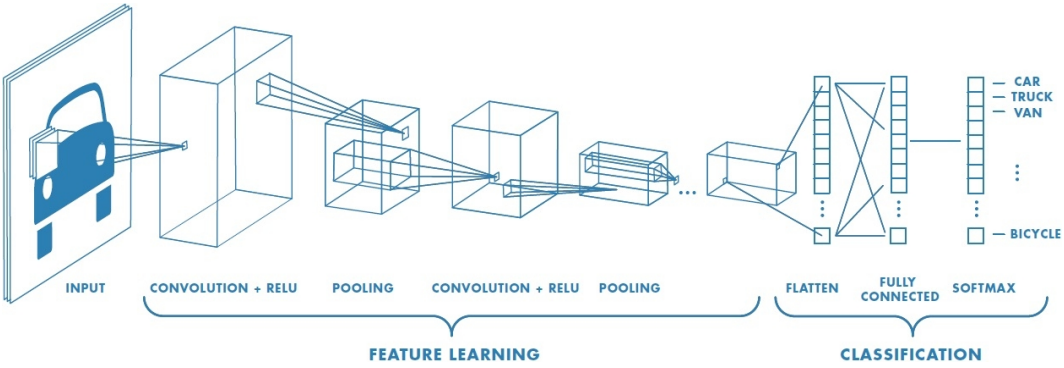


FIGURE 2: A general CNN architecture which is merges the convolution layer and activation layer into one (Mathworks, 2016)

Références

ACHARD F., EVA H. D., STIBIG H.-J., MAYAUX P., GALLEGO J., RICHARDS T. & MALINGREAU J.-P. (2002). Determination of deforestation rates of the world's humid tropical forests. *Science*, **297**(5583), 999–1002.

BABENKO A., SLESAREV A., CHIGORIN A. & LEMPITSKY V. (2014). Neural codes for image retrieval. In *European conference on computer vision*, p. 584–599 : Springer.

BARRETO T. L., ROSA R. A., WIMMER C., MOREIRA J. R., BINS L. S., CAPPABIANCO F. A. M. & ALMEIDA J. (2016). Classification of Detected Changes From Multitemporal High-Res Xband SAR Images : Intensity and Texture Descriptors From SuperPixels. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **9**(12), 5436–5448.

COLLINS J. B. & WOODCOCK C. E. (1996). An assessment of several linear change detection techniques for mapping forest mortality using multitemporal landsat tm data. *Remote sensing of environment*, **56**(1), 66–77.

GEOGRAPHIC N. (2017). Climate 101 : Deforestation. https://www.nationalgeographic.com/environment/global-warming/deforestation/.

GUO Y., LIU Y., OERLEMANS A., LAO S., WU S. & LEW M. S. (2016). Deep learning for visual understanding : A review. *Neurocomputing*, **187**, 27–48.

HE K., SUN J. & TANG X. (2011). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(12), 2341–2353.

HIRSCHMUGL M., GALLAUN H., DEES M., DATTA P., DEUTSCHER J., KOUTSIAS N. & SCHARDT M. (2017). Methods for mapping forest disturbance and degradation from optical earth observation data : a review. *Current Forestry Reports*, **3**(1), 32–45.

HU F., XIA G.-S., HU J. & ZHANG L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, **7**(11), 14680–14707.

IONESCU B., MÜLLER H., VILLEGAS M., ARENAS H., BOATO G., DANG-NGUYEN D.-T., CID Y. D., EICKHOFF C., DE HERRERA A. G. S., GURRIN C., ISLAM B., KOVALEV V., LIAUCHUK V., MOTHE J., PIRAS L., RIEGLER M. & SCHWALL I. (2017). Overview of imageclef 2017 : Information extraction from images. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, p. 315–337 : Springer.

JOSHI N., BAUMANN M., EHAMMER A., FENSHOLT R., GROGAN K., HOSTERT P., JEPSEN M. R., KUEMMERLE T., MEYFROIDT P., MITCHARD E. T. A., REICHE J., RYAN C. M. & WASKE B. (2016). A review of the application of optical and radar remote sensing data fusion to land use mapping and monitoring. *Remote Sensing*, **8**(1).

KAGGLE I. (2017). Planet : Understanding the amazon from space | kaggle. https://www.kaggle.com/c/planet-understanding-the-amazon-from-space. Accessed : 2018-01-25.

KHAN S. H., HE X., PORIKLI F. & BENNAMOUN M. (2017). Forest change detection in incomplete satellite images with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, **55**(9), 5407–5423.

KRIZHEVSKY A., SUTSKEVER I. & HINTON G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, p. 1097–1105.

KUSSUL N., LAVRENIUK M., SKAKUN S. & SHELESTOV A. (2017). Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters*, **14**(5), 778–782.

LAGRANGE A., SAUX B. L., BEAUPÈRE A., BOULCH A., CHAN-HON-TONG A., HERBIN S., RANDRIANARIVO H. & FERECATU M. (2015). Benchmarking classification of earth-observation data : From learning explicit features to convolutional networks. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, p. 4173–4176.

LECUN Y., BOTTOU L., BENGIO Y. & HAFFNER P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

LU D., MAUSEL P., BRONDIZIO E. & MORAN E. (2004). Change detection techniques. *International journal of remote sensing*, **25**(12), 2365–2401.

MATHWORKS (2016). Convolutional neural network. https://fr.mathworks.com/discovery/convolutional-neural-network.html.

MERTENS B. & LAMBIN E. F. (1997). Spatial modelling of deforestation in southern cameroon : spatial disaggregation of diverse deforestation processes. *Applied Geography*, **17**(2), 143–162.

MILLER L. D. L. D., NUALCHAWEE K., TOM C. C. & CENTER G. S. F. (1978). *Analysis of the dynamics of shifting cultivation in the tropical forests of northern Thailand using landscape modeling and classification of Landsat imagery*. Greenbelt, Md. : National Aeronautics and Space Administration, Goddard Space Flight Center. "May 1978.".

MOTHE J., NY HOAVY N. & RANDRIANARIVONY M. I. (2017). IRIT & MISA at Image CLEF 2017 - Multi label classification. In *International Conference of the CLEF Association, CLEF 2017 Labs Working Notes*, volume 1866 of *ISSN 1613-0073* : CEUR Workshop Proceedings.

MÜLLER H., GRIFFITHS P. & HOSTERT P. (2016). Long-term deforestation dynamics in the brazilian amazon—uncovering historic frontier development along the cuiabá–santarém highway. *International Journal of Applied Earth Observation and Geoinformation*, **44**, 61–69.

O'CONNOR B., SECADES C., PENNER J., SONNENSCHEIN R., SKIDMORE A., BURGESS N. D. & HUTTON J. M. (2015). Earth observation as a tool for tracking progress towards the aichi biodiversity targets. *Remote sensing in ecology and conservation*, **1**(1), 19–28.

OLOFSSON P., HOLDEN C. E., BULLOCK E. L. & WOODCOCK C. E. (2016). Time series analysis of satellite data reveals continuous deforestation of new england since the 1980s. *Environmental Research Letters*, **11**(6), 064002.

POHL C. & VAN GENDEREN J. L. (1998). Review article multisensor image fusion in remote sensing : concepts, methods and applications. *International journal of remote sensing*, **19**(5), 823–854.

REICHE J., HAMUNYELA E., VERBESSELT J., HOEKMAN D. & HEROLD M. (2018). Improving near-real time deforestation monitoring in tropical dry forests by combining dense sentinel-1 time series with landsat and alos-2 palsar-2. *Remote Sensing of Environment*, **204**, 147 – 161.

REICHE J., VERBESSELT J., HOEKMAN D. & HEROLD M. (2015). Fusing landsat and sar time series to detect deforestation in the tropics. *Remote Sensing of Environment*, **156**, 276–293.

SALBERG A. B. (2015). Detection of seals in remote sensing images using features extracted from deep convolutional neural networks. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, p. 1893–1896.

SCHMIDT M., LUCAS R., BUNTING P., VERBESSELT J. & ARMSTON J. (2015). Multi-resolution time series imagery for forest disturbance and regrowth monitoring in queensland, australia. *Remote Sensing of Environment*, **158**, 156–168.

SCOTT K. (2017). Forest recognition : Planet launches kaggle competition. `https://www.planet.com/pulse/forest-recognition-planet-launches-kaggle-competition/`. Accessed : 2018-01-25.

SHIMABUKURO Y. E., BATISTA G., MELLO E., MOREIRA J. & DUARTE V. (1998). Using shade fraction image segmentation to evaluate deforestation in landsat thematic mapper images of the amazon region. *International Journal of Remote Sensing*, **19**(3), 535–541.

SINGH A. (1989). Review article digital change detection techniques using remotely-sensed data. *International journal of remote sensing*, **10**(6), 989–1003.

TARIQ M. & AZIZ R. (2015). An overview of deforestation causes and its environmental hazards in khyber pukhtunkhwa. *Journal of Natural Sciences Research*, **5**(1).

WITTEN I. H., FRANK E., HALL M. A. & PAL C. J. (2016). *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann.

ZHANG C., PAN X., LI H., GARDINER A., SARGENT I., HARE J. & ATKINSON P. M. (2017). A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*.

ZHANG J. (2010). Multi-source remote sensing data fusion : status and trends. *International Journal of Image and Data Fusion*, **1**(1), 5–24.

Détection d'influenceurs dans des médias sociaux

Kévin Deturck[1, 2]
(1) ERTIM, 2 rue de Lille, 75007 Paris, France
(2) Viseo, 4 avenue Doyen louis Weil, 38000 Grenoble, France
kevin.deturck@viseo.com

RÉSUMÉ

Les influenceurs ont la capacité d'avoir un impact sur d'autres individus lorsqu'ils interagissent avec eux. Détecter les influenceurs permet d'identifier les quelques individus à cibler pour toucher largement un réseau. Il est possible d'analyser les interactions dans un média social du point de vue de leur structure ou de leur contenu. Dans nos travaux de thèse, nous abordons ces deux aspects. Nous présentons d'abord une évaluation de différentes mesures de centralité sur la structure d'interactions extraites de Twitter puis nous analysons l'impact de la taille du graphe de suivi sur la performance de mesures de centralité. Nous abordons l'aspect linguistique pour identifier le changement d'avis comme un effet de l'influence depuis les messages d'un forum.

ABSTRACT

Influencer detection in social medias

There is an increasing interest in the detection of influencers on social medias. Different features may be used: the text of the messages and the structure of the network. The initial PhD works presented in this paper explore these two aspects. We evaluate the effectiveness of centrality measures from the state of the art in detecting Twitter influencers building graphs from interactions between Twitter users. In a second experimentation, we analyze the impact of the network size on the selection of the most appropriate centrality algorithms. We segment Twitter users according to the number of their followers and build their respective underlying "following graph". We then run the selected algorithms on the different graphs and evaluate their performance throughout the different graph sizes. We finally present a current work on linguistic features to detect opinion change as an influence effect.

MOTS-CLÉS : influence, média social, réseau, centralité, opinion

KEYWORDS: influence, social media, network, centrality, opinion

1 Introduction

Nos travaux de thèse visent à détecter les individus ayant une influence dans des médias sociaux. Nous entendons par "influence" le pouvoir que possède un individu qui parvient à mobiliser d'autres individus en faveur d'une action ou d'une opinion.

Détecter automatiquement les influenceurs dans des médias sociaux peut être vu comme une tâche pour un système de recherche d'information destiné à un utilisateur qui aurait besoin de connaitre les personnes meneuses et leurs propos dans un certain domaine d'activité. Les influenceurs constituent

aussi des points d'entrée efficaces pour la diffusion ciblée d'une information lors de campagnes de santé publique, pour la promotion d'un produit ou encore la gestion de réputation en ligne.

La pluralité des médias sociaux et la variété des informations qu'ils contiennent constituent autant de facettes possibles pour caractériser l'influence ouvrant ainsi de larges perspectives à l'élaboration d'un cadre formel permettant de la détecter. Il s'agit aussi d'une complexité supplémentaire pour modéliser l'information utile à la détection des influenceurs. Nous aurons à distinguer et à recouper la nature des informations disponibles à travers les différents médias sociaux pour les intégrer dans notre modèle. Par exemple, comparer la portée des interactions "Favori" dans Twitter et "J'aime" dans Facebook portant toutes les deux sur un contenu. Aussi, les textes que nous étudierons à travers les différents médias sociaux auront des formats tout à fait différents qui constitueront un corpus particulièrement hétérogène. Par exemple, nous ne pourrons pas aborder un texte de forum avec les mêmes modalités que celles utilisées pour un Tweet.

Les interactions entre les individus sont à la base de toute influence. Dans un média social, ces interactions peuvent être analysées d'après leur structure ou leur contenu. La structure peut porter une signification variée qui dépend à la fois des natures d'interaction considérées et de l'interprétation que nous leur prêtons. Nous nous situons dans le domaine du Traitement Automatique des Langues et nous focalisons donc nos travaux sur le contenu textuel des médias sociaux même si d'autres types de contenu comme l'image peuvent être utilisés par les influenceurs.

Identifier structurellement les influenceurs dans un groupe d'individus revient à trouver les clés de voûte de leurs interactions. Pour ce faire, nous devons trouver une représentation des interactions qui tienne compte de leurs natures variées. Par exemple, nous devons choisir comment inclure dans une même structure les interactions Twitter d'abonnement et de Retweet afin qu'elles soient les plus significatives possibles.

Nous avons à déterminer les phénomènes linguistiques qui permettent, du point de vue des influenceurs, de mettre en œuvre leur pouvoir et qui, du point de vue des individus cibles, font état de leurs réactions face aux influenceurs. Le principe est alors d'extraire des régularités expliquant sous différents types éventuels le phénomène d'influence par la langue écrite.

Les messages que nous devons analyser contiennent une expression informelle, un défi intéressant pour l'analyse linguistique parce que hors d'une grammaire standard de la langue. Pour l'analyse automatique, nous devrons mettre en œuvre un processus important de prétraitement permettant de ramener les textes des médias sociaux à une langue standard qui pourra être traitée par des outils de Traitement Automatique de la Langue. Le caractère elliptique des messages, en particulier dans les réseaux sociaux, peut empêcher l'identification de phénomènes linguistiques tels qu'ils sont traditionnellement décrits. Cela requiert une nouvelle modélisation des phénomènes qu'on veut repérer pour qu'ils soient applicables aux modes d'expression récents que nous souhaitons analyser. Prenons l'exemple d'un influenceur qui argumente pour changer l'opinion de quelqu'un. Pour caractériser son argumentation à l'intérieur d'un Tweet et la reconnaitre dans d'autres messages, nous ne pourrons certainement pas utiliser exactement les modèles traditionnels de l'argumentation.

2 État de l'art

2.1 Sur la détection structurelle des influenceurs

La centralité permet de mesurer l'importance structurelle d'un nœud dans un graphe. Nous pouvons analyser un média social par la structure de ses interactions représentées comme les arcs d'un graphe. Dans un réseau, l'influenceur est un utilisateur qui polarise les interactions, c'est donc un nœud central du graphe correspondant.

Le positionnement d'un individu dans un réseau est déterminé par ses connexions avec d'autres utilisateurs. Ces connexions sont données par les interactions de tous types entre les individus d'un média social. C'est plus particulièrement les réseaux sociaux qui sont à l'étude ici puisque la nature même du réseau rend essentielle l'analyse de ses liens. Un influenceur est alors identifié par un positionnement central à l'intérieur du réseau. Différentes mesures de centralité ont été proposées donnant lieu à des interprétations d'influence différentes. (Mariani et al., 2014) analysent les réseaux de citations pour mesurer l'influence scientifique. Ils appliquent une mesure de centralité qui prend en compte la proximité d'un utilisateur par rapport aux autres dans le graphe selon des liens que sont les citations des publications et les collaborations. La valeur de centralité pour chaque utilisateur indique son degré d'influence. (Sheikhahmadi et al., 2017) mettent en avant l'importance de la communauté dans les interactions des utilisateurs d'un réseau social. Les auteurs utilisent la tendance des utilisateurs à communiquer à l'intérieur d'un groupe constitué par exemple autour d'une thématique. Ils affirment donc que les influenceurs doivent être identifiés pour chaque communauté et divisent donc le graphe par détection de communauté avant d'utiliser la quantité d'interactions pour pondérer l'influence de chaque utilisateur dans sa communauté. (Khadangi, Bagheri, 2017) distinguent les interactions marquant une affinité entre les utilisateurs de celles qui reflètent l'activité propre de d'un utilisateur comme les « J'aime » sur Facebook, ces dernières étant moins étudiées pour l'influence.

Puisque les influenceurs ont la capacité de mobiliser d'autres individus pour des actions ou des opinions, nous pouvons aussi les identifier en analysant les dynamiques dans la structure des interactions. Un influenceur est alors un individu qui parvient à propager une attitude le plus rapidement et le plus longuement à travers un réseau. (Dave et al., 2011) cherchent à prédire quelle sera la dynamique de propagation d'une action à partir d'un certain utilisateur pour prédire son influence. (Jabeur et al., 2012) voient plus généralement les influenceurs comme ayant une capacité à propager une information. Les auteurs se concentrent ainsi, dans Twitter, sur les Retweets pour construire un graphe d'utilisateurs qu'ils analysent à la manière de Page Rank en considérant l'importance des utilisateurs Retweetant pour calculer l'influence de l'utilisateur Retweeté. (Gionis et al., 2013) s'intéressent à l'opinion des utilisateurs pour prédire ceux dont elle va le mieux se maximiser à travers un réseau d'après la constitution de leur voisinage respectif. Ces utilisateurs sont les clés du problème de maximisation d'influence à travers un réseau social. Le principe est de trouver l'ensemble d'utilisateurs qui permettra de diffuser une information le plus rapidement possible dans un réseau.

2.2 Sur la détection linguistique des influenceurs

Les approches qui analysent les textes issus de médias sociaux pour détecter les influenceurs, comme (Biran et al., 2012), essayent d'identifier les marqueurs d'un discours influent. L'influenceur

s'exprime souvent pour soutenir une revendication à propos d'un certain sujet. Cette revendication contient un point de vue, elle se caractérise donc essentiellement par une énonciation subjective. À ce propos, (Pak, Paroubek, 2010) cherchent à repérer automatiquement le point de vue adopté dans des Tweets par apprentissage automatique sur les catégories morphosyntaxiques en présence. Les auteurs mettent en avant l'utilisation d'adjectifs superlatifs pour renforcer l'expression des émotions et la forte présence de pronoms personnels indiquant une personnalisation du discours comme des traits particulièrement discriminants pour le discours subjectif. En général, l'influenceur essaye de faire adhérer d'autres personnes à son discours en utilisant deux grands procédés argumentatifs : persuader et convaincre.

La persuasion se caractérise par une implication fortement personnelle qui s'exprime avec des émotions et des opinions. La richesse des émotions exprimées peut, dans le cas de la persuasion, compenser l'absence de raisonnement. Persuader fait appel à une argumentation intuitive. La répétition d'un même propos entre ainsi dans le cadre de la persuasion.

Par contraste avec la persuasion, convaincre nécessite le développement d'une argumentation plus rationnelle pour servir la revendication exprimée. Il s'agit alors d'un discours moins direct fondé sur le raisonnement. (Rosenthal, 2014)

Détecter un texte argumentatif revient à faire de l'analyse du discours. L'argumentation est décrite en certaines relations rhétoriques présentées dans (Biran & Rambow, 2011) qui lient une partie revendicative à une partie justificative. Chacune de ces parties constitue une unité de discours à identifier. Une fois les unités de discours élémentaires identifiées, il est question de trouver les relations rhétoriques qu'il y a entre elles (Danlos, 2011). Ces relations permettent de structurer l'argumentation.

(Quercia et al., 2011) s'intéressent à l'usage de la langue chez les utilisateurs de Twitter dans une approche statistique pour identifier les influenceurs. Ils mettent en avant l'usage d'un champ lexical de l'émotion pour instaurer une intimité avec les lecteurs et ainsi faciliter l'impact du message à transmettre. Cette intimité est complétée par un usage de la troisième personne pour donner aux lecteurs l'impression d'appartenance à une même communauté qui se démarque des autres.

Pour repérer l'évolution d'une opinion sur Twitter, (Bifet et al., 2011) analysent la spécificité d'utilisation des termes en présence pour chaque Tweet d'un utilisateur à travers des fenêtres temporelles. Ils considèrent la polarité positive ou négative du lexique pour attribuer une valeur à l'opinion et ils utilisent les Hashtags pour créer des ensembles de Tweets portant sur une même thématique.

L'analyse linguistique peut aussi servir à orienter la détection d'influenceurs vers une thématique, nous avons mentionné précédemment l'importance de la communauté, notamment thématique, pour l'exercice d'une influence. (Hamzehei et al., 2017) analysent l'influence des utilisateurs de réseaux sociaux à l'aune des topiques détectés dans leurs messages.

3 Expérimentations réalisées

Nous avons, en section 1, défini l'influence comme un pouvoir, ce qui nous amène à caractériser les influenceurs selon deux aspects : les ressources qui leur permettent d'influencer et les effets de leur influence du point de vue des individus cibles. Nous allons rechercher les ressources comme les

effets des influenceurs aussi bien dans le contenu des messages échangés entre les influenceurs et leur audience que sur des caractéristiques topologiques concernant la structure et la nature de leurs interactions. Nos expérimentations comportent ainsi deux pans avec la relations d'un individu Nous en déduisons de nouvelles pistes afin de créer un système hybride entre les aspects de structure et de contenu souvent étudiés indépendamment.

3.1 Comparaison de mesures de centralité pour la détection d'influenceurs Twitter

Le but de cette expérimentation est d'évaluer des algorithmes mesurant chacun une certaine centralité. Nous pouvons distinguer deux types de centralité. La centralité locale qui prend en compte uniquement le voisinage d'un nœud pour lui attribuer une valeur de centralité. La centralité globale qui regarde le nœud par rapport à l'ensemble du graphe afin d'évaluer sa centralité. Pour chaque mesure de centralité, nous évaluons sa capacité à modéliser l'influence des individus représentés comme les nœuds d'un graphe qui représente la structure de leurs interactions.

Nous présentons six mesures de centralité qui nous permettent d'évaluer autant de types de centralité.

- **Degré entrant** : c'est une mesure de centralité locale puisqu'elle n'est basée que sur le calcul du nombre de liens entrants pour un nœud donné. En termes d'influence, c'est une polarisation directe envers un utilisateur (Freeman, 1978)

- **Intermédiarité** et **Proximité** : il n'y a pas de résultat pour Intermédiarité et Proximité parce qu'ils ont besoin de calculer les chemins les plus courts entre tous les nœuds du graphe, ce qui nécessite un graphe connecté, ce n'est pas le cas des graphes que nous avons extraits (Freeman, 1978) Nous les mentionnons tout de même parce qu'elles sont des mesures importantes qui ont bien été prises en compte dans nos expérimentations.

- **Hits** : apporte une sémantique supplémentaire à la centralité globale avec une distinction mutuelle entre un score d'autorité et un score de relais. L'autorité est importante parce que particulièrement écoutée par des relais qui sont eux-mêmes importants parce qu'ils écoutent beaucoup d'autorités (Kleinberg, 1999)

- **Page Rank** : calcule l'importance d'un nœud dans un graphe d'après la valeur de chaque lien pointant vers lui et l'importance du nœud à l'origine, avec une initialisation uniforme des poids et par itérations successives sur tous les nœuds du graphe jusqu'à ce que le poids de chacun soit à l'équilibre, cet algorithme donne des valeurs de centralité globale. La valeur de chaque se dilue avec le nombre de liens sortants de son nœud d'origine. Est ajoutée une probabilité uniforme pour tous les nœuds du graphe de se départir de la structure du réseau pour « sauter » d'un nœud à l'autre (Page et al., 1999)

- **Leader Rank** : entend améliorer la modélisation de Page Rank pour les réseaux sociaux en affirmant que la probabilité de passage d'un nœud à un autre sans utiliser les arcs du graphe ne doit pas être uniforme mais inversement proportionnelle au nombre de liens sortants qui sont disponibles

Ces algorithmes attribuent à chacun des nœuds du graphe un score de centralité permettant d'établir un classement d'utilisateurs pour un réseau considéré. Pour les évaluer, nous devons mesurer la qualité du classement produit à partir de chaque algorithme en fonction d'une référence qui donne l'influence de chaque utilisateur. Nous avons ajouté une Baseline fondée sur un classement au hasard des utilisateurs pour mieux appréhender les résultats des algorithmes évalués.

Comme référence, nous avons choisi le jeu de données conçu pour la compétition RepLab 2014 (Amigó et al., 2014). Cette compétition proposait notamment une tâche consistant à classer des utilisateurs Twitter du plus influent au moins influent. Le jeu de données contient plus de 7000 comptes Twitter catégorisés selon qu'ils appartiennent au domaine de la banque, de l'automobile ou à des domaines différents des deux précédents.. Ils ont été annotés par les spécialistes de la réputation en ligne Llorente & Cuenca[1]. Cette annotation considère l'influence réelle (dans le monde) des utilisateurs, en indiquant s'ils sont influenceurs ou pas. Le jeu de données contient en moyenne 1/3 d'influenceurs. Nous avons construit un échantillon de 50 utilisateurs du domaine de la banque pour limiter la taille du graphe à construire et le temps d'extraction des informations depuis l'API Twitter[2], qui impose des limites à la quantité d'information extraite par intervalle de temps. Nous avons fait en sorte de conserver la proportion d'influenceurs originale (1/3) pour garder une certaine comparabilité avec les systèmes de la compétition.

À partir de cet ensemble d'utilisateurs initial, nous extrayons deux types d'interaction.

- Le **suivi** qui constitue une audience et donc un terreau d'utilisateurs pour l'exercice d'une influence

- Le **Retweet** ou la rediffusion d'un contenu constituant une réaction comme un effet d'influence

À partir des utilisateurs initiaux, nous extrayons ces deux types d'interaction et donc de nouveaux utilisateurs qui seront représentés dans un graphe pour chacun des deux types d'interaction.

Nous comparons la référence binaire aux classements issus des algorithmes avec la mesure Mean Average Precision (MAP) utilisée pour RepLab 2014. Elle est fondée sur l'intuition en Recherche d'Information que les résultats les plus pertinents, les influenceurs, doivent apparaitre au début. Nous avons calculé MAP par la formule qui suit :

$$MAP = \frac{1}{n} \sum_{i=1}^{N} p(i) R(i)$$

avec N le nombre total d'utilisateur, n le nombre d'influenceurs correctement trouvés, $p(i)$ la précision au rang i (en ne considérant que les i premiers utilisateurs trouvés) et $R(i)$ est à 1 si l'utilisateur au rang i est un influenceur sinon à 0.

[1] www.llorenteycuenca.com/en/
[2] www.developer.twitter.com

Sur un graphe de suivi

Nombre de nœuds	5,067,480
Nombre d'arcs	5,149,491
Densité	10^{-7}

Tableau 1 : caractéristiques du graphe de suivi construit

Algorithme	Baseline	Degré entrant	Page Rank	Leader Rank	Hits
MAP (%)	**38,67**	**43,49**	**44,28**	**44,53**	**51,68**

Tableau 2 : résultats des mesures de centralité sur le graphe de suivi

- On observe *Tableau 1* une faible densité du graphe, les utilisateurs du corpus initial se suivent peu, ce qui explique pourquoi Page Rank et Leader Rank, mesures de centralité globales, donnent *Tableau 2* des résultats similaires au degré entrant, locale

- Hits se démarque en distinguant les influenceurs comme étant des « autorités », ce qui permet de donner comme première caractéristique des influenceurs d'être suivis par des « relais » tels que nous les avons précédemment présentés.

Sur un graphe de Retweet

Nombre de nœuds	2099
Nombre d'arcs	2051
Densité	10^{-4}

Tableau 3 : caractéristiques du graphe de Retweet construit

Algorithme	Baseline	Degré entrant	Page Rank	Leader Rank	Hits
MAP (%)	**38,67**	**40,91**	**40,91**	**40,91**	**40,91**

Tableau 4 : résultats des mesures de centralité sur le graphe de Retweet

- Même problème de connectivité que sur le graphe de suivi pour expliquer l'absence de résultat d'Intermédiarité et de Proximité

- À nouveau, la faible densité du graphe, signifiant ici que les utilisateurs se retweetent peu, empêche les mesures de centralité globales d'apporter de meilleurs résultats par rapport au degré entrant, local, qui ne prend en compte que le voisinage direct

- Le fait que Hits ne se distingue pas sur le graphe de Retweet peut indiquer que l'information de suivi est plus significative pour détecter les influenceurs.

Algorithme	UTDBRG	Lys	LIA	UAMCLYR	ORM_UNED
MAP	**0,41**	**0,52**	**0,45**	**0,49**	**0,32**

Tableau 5 : Résultats des système de RepLab sur la banque

Pour information, nous ajoutons *Tableau 5* les résultats des systèmes ayant participé à la compétition RepLab et diffusés dans (Amigó et al., 2014). Nous avons pris le meilleur essai pour chaque système sur l'ensemble du corpus et sur le même domaine que notre étude. Les meilleurs systèmes ont surtout utilisé des métadonnées des profils comme le texte de présentation, le nombre de suiveurs (Lys, UAMCLYR), la présence d'une image de profil et le statut de vérification du compte (Lys).

3.2 Étude de l'impact du nombre de suiveurs pour distinguer des influenceurs Twitter par des mesures de centralité

Nous nous concentrons pour cette expérimentation sur l'information de suivi qui nous a semblé plus significative. Nous restreignons notre sélection de mesures de centralité à Page Rank, Leader Rank et Hits qui n'ont pas besoin d'un graphe connexe pour calculer un résultat. Ces trois algorithmes opèrent itérativement pour calculer un score approximé tolérant pour la convergence une certaine variation entre deux itérations. Nous cherchons à la fois à analyser le comportement des mesures de centralité en faisant évoluer la taille du graphe de suivi et aussi à déterminer dans quelle mesure le nombre de suiveurs est significatif en l'utilisant dans des intervalles de valeur différents.

Nous utilisons le même corpus que précédemment en créant cette fois-ci des échantillons de 50 utilisateurs suivant les intervalles de nombre de suiveurs : [1000-5000], [5000-10,000], [10,000-50,000], [50,000-100,000], [100,000-500,000]. Nous faisons en sorte que chaque échantillon comprenne la même proportion d'influenceurs : 1/3 (proportion d'influenceurs du corpus d'origine). Nous construisons un graphe de suivi pour chacun des segments ce qui donne cinq graphes de suivi aux tailles différentes sur lesquelles nous appliquons les mesures de centralité précédemment sélectionnées.

Segments/Caractéristiques	**[1000-5000]**	**[5000-10,000]**	**[10,000-50,000]**	**[50,000-100,000]**	**[100,000-500,000]**
Nombre de nœuds	122,060	361,422	1,000,180	3,156,671	9,708,482
Nombre d'arcs	124,747	372,939	1,034,110	3,322,007	10,521,923
Densité	10^{-6}	10^{-6}	10^{-6}	10^{-7}	10^{-7}

Tableau 6 : Caractéristiques des graphes de suivi selon les segments

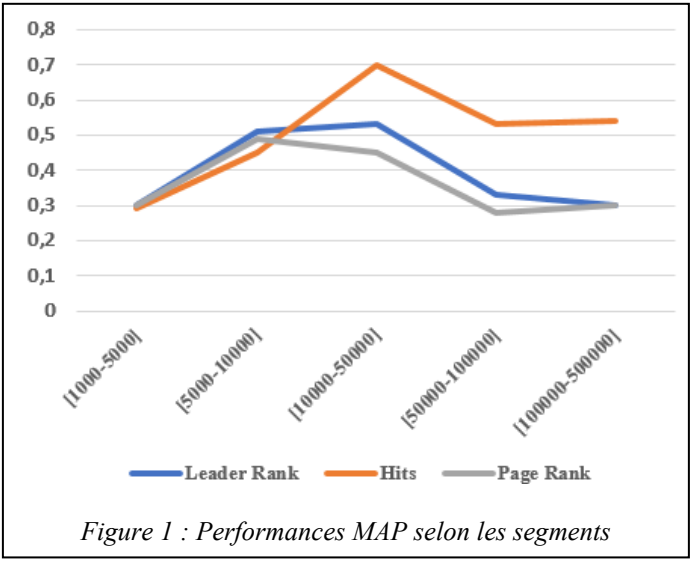


Figure 1 : Performances MAP selon les segments

La taille des segments augmentant, l'écart possible en nombre de suiveurs entre les utilisateurs augmente aussi, ce qui accroît potentiellement le pouvoir discriminant de ce critère pour distinguer les influenceurs. Le fait que la performance des trois algorithmes augmente jusqu'à un certain point avec la taille des segments montre que l'information de suivi est effectivement significative pour la détection des influenceurs. Nous expliquons globalement la baisse des performances en disant que le taille du graphe devient trop importante (*Tableau 3*, facteur 3 d'agrandissement du graphe entre deux segments) pour que l'approximation calculée par les trois algorithmes suffise à représenter correctement les forces en présence dans tout le graphe.

Nous constatons *Figure* 1 que deux algorithmes sont plus stables : Page Rank et Leader Rank. Leur comportement similaire peut s'expliquer par le fait que le second est une modification du premier. Aussi, ces deux algorithmes ont la particularité d'autoriser le fait de ne pas suivre la structure du graphe pour passer d'un nœud à l'autre, cela explique pourquoi ils sont moins sensibles aux variations de taille du graphe.

Pour Hits, la dynamique est plus forte (50% de progression relative entre [1000-5000] et [10,000-50,000] et 15% de baisse absolue entre [10,000-50,000] et [100,000-500,000]). Contrairement aux deux algorithmes précédents, Hits suit strictement la structure du graphe : pas de « saut » d'un nœud à l'autre ni de dilution de la valeur des liens. Il est donc plus sensible à la modification de la taille du graphe. Le fait que l'augmentation de marge en nombre de suiveurs entre les utilisateurs aide l'algorithme qui se base le plus sur la structure montre aussi la significativité de l'information de suivi.

La modélisation « nœud de terre » dans Leader Rank tient à mieux modéliser le comportement des utilisateurs d'un réseau social lorsque le nombre de liens sortants augmente et donc lorsque la taille du graphe augmente. Or, nous pouvons constater *Figure 1* que Leader Rank se détache positivement de Page Rank lorsque la taille des segments croît. C'est un signal que la modélisation de Leader Rank est effectivement meilleure que celle de Page Rank.

3.3 Détection linguistique du changement d'avis comme un effet de l'influence dans un corpus "Change My View[3]"

Notre parti pris est de détecter les influenceurs par leur influence effective là où des travaux précédents émettent des hypothèses sur l'influençabilité de critères linguistiques. Ainsi, nous voyons la détection du changement d'avis comme une première étape à la détection d'influenceurs en tant qu'un effet d'influence. Il s'agirait pour la suite de repérer la source du changement d'avis pour identifier l'influenceur. L'objectif est d'obtenir des influenceurs réels plutôt que des potentiels influenceurs.

Corpus/Caractéristiques	#fils de discussion Pour l'entrainement	#messages Pour l'entrainement	#fils de discussion Pour l'évaluation	#messages Pour l'évaluation
Initial	18,363	1,114,533	2,263	145,733
Pré-filtré	10,743	128,901	1,529	20,883
Final	3,191	42,776	672	9463

Tableau 7 : Statistiques générales des corpus jusqu'au filtrage final

Nous avions besoin d'une ressource contenant des manifestations textuelles de changement d'avis. Nous avons choisi le forum en anglais « Change My View », précédemment étudié pour des tâches connexes par (Tan et al., 2016). Le principe de ce forum est que l'auteur initial d'un fil de discussion expose son point de vue sur une thématique puis il demande aux lecteurs de le faire changer d'avis. Les autres participants au fil de discussion vont alors faire en sorte de développer une argumentation qui contredise assez bien le point de vue de l'auteur initial pour modifier son point de vue. Lorsque l'auteur initial reconnait qu'un message a eu l'effet escompté, il lui attribue un « delta », ce qui revient à une annotation « ad hoc » des messages gagnants. C'est cette annotation que nous utiliserons comme référence. L'auteur initial doit en plus justifier cette récompense dans un message qui explicite son changement d'avis.

Nous utilisons le corpus extrait par (Tan et al., 2016) qui essayent notamment de prédire les arguments qui vont créer un changement d'avis. Toujours dans l'optique de détecter un influenceur effectif, nous adoptons une approche inverse puisque nous partons des réponses à ces arguments pour détecter ceux qui ont eu un impact. Les auteurs ont extrait plus de 20,000 fils de discussion depuis la création du forum en 2013 jusqu'en 2015 (les statistiques du corpus initial sont dans le *Tableau 6*). Cela donne plus de 1,000,000 de messages pour environ 80,000 participants uniques, donc beaucoup d'habitués. Le corpus est déjà divisé entre une partie pour l'entrainement (90%) et une partie pour l'évaluation (10%). Nous partons du corpus que les auteurs ont filtré pour leur tâche sur la résistance à la persuasion (statistiques générales du corpus pré-filtré en *Tableau 6*). Ils posent des contraintes de pertinence sur une participation minimale pour l'auteur initial et pour les autres à l'intérieur d'un fil de discussion. Puisque nous cherchons à détecter le changement d'avis pour les auteurs initiaux, nous analysons leurs messages seulement dans les fils de discussion où ils changent d'avis au moins une fois (les statistiques générales du corpus final en *Tableau 6*). Les discussions qui contiennent au moins un changement d'avis représentent 30% du corpus pré-filtré par les auteurs (*Tableau 6*). Dans le corpus final, le taux d'exemples positifs (messages d'un auteur initial exprimant un changement d'avis) est de 10%, ce qui réduit légèrement le déséquilibre entre les

[3] www.reddit.com/r/changemyview/

classes (2% initialement). Tous les messages sont analysés sans distinction d'auteur. Nous avons supprimé des messages les marques liées à l'attribution d'une récompense comme « delta » pour ne pas biaiser la classification.

Nous pouvons voir ce travail comme une tâche de classification binaire puisque pour chaque message, nous devons dire s'il exprime un changement d'avis ou pas. Nous utilisons un classifieur par régression logistique qui convient particulièrement à ce genre de classification. Le corpus original comme notre échantillon d'entrainement contiennent seulement 2% de messages exprimant un changement d'avis. Cela constitue un biais pour l'apprentissage du classifieur qui pourrait simplement annoter les messages avec la classe majoritaire (pas de changement d'avis) afin d'obtenir un résultat correct. Pour contrebalancer ce biais, nous mesurons la performance de notre classifieur en utilisant la mesure Area Under ROC Curve qui prend en compte le taux de vrais positifs par rapport au taux de faux positifs.

Dans cette expérimentation, toujours en cours, nous avons commencé par déterminer les descripteurs les plus pertinents en les utilisant séparément. Dans la poursuite de nos travaux, nous travaillerons sur la combinaison de ces traits pour obtenir le meilleur résultat possible.

Descripteur	Nombre de mots (référentiel)	Sac de mots	POS	Style	Passé
Score AUC (%)	**51,38**	**82,11**	**60,97**	**64,70**	**57,15**

Tableau 8 : Résultats de descripteurs pour la détection de changement d'avis

Le trait le plus simple, qui constitue le référentiel de notre évaluation, consiste à utiliser le nombre de mots du message. Nous obtenons un résultat assez neutre pour un référentiel à 51,38% (cf. *Tableau 8*).

Nous nous sommes demandé s'il y avait un emploi de termes particulier pour les messages exprimant un changement d'avis. Nous utilisons une représentation en sac de mots des tokens présents dans chaque message. Nous obtenons le meilleur résultat avec 82,11% pour ce trait seul (*Tableau 8*). Nous avons analysé les termes les plus discriminants pour donner du sens à ce résultat chiffré. Pour la classe positive (changement d'avis), le terme au poids le plus important (10,4) est « convinced », qui est central pour exprimer l'impact d'un argument gagnant comme dans « This one finally convinced me » (message *t1_cna7wg7*). La classe positive contient des termes de concession comme « concede », « still » qui marquent un tournant. Nous observons aussi un poids très important (8,1) pour « hadn » qui semble marquer une remise en cause du point de vue passé comme dans « I hadn't considered the obvious fact that (…) » (message *t1_cnbkumq*). Toujours dans ce modèle avec un terme au passé à polarité négative, nous observons l'émergence du terme « forgot » (6,5). Enfin, il y a un ensemble de termes ayant trait à la clairvoyance tels que « guesss » ou « realize » (7), qui dénotent la compréhension d'une nouvelle vision des choses comme dans « I think this post really helped me realize (…) » (message *t1_cni1gsy*). Pour la classe négative (messages sans expression d'un changement d'avis), le terme le plus fort (4,7) est « talking », qui peut à la fois montrer que la discussion n'est pas résolue et qu'il y a un malentendu comme dans « I am talking about (…) » (message *t1_cngt9pv*). Ce malentendu transparait aussi dans les messages de la classe négative avec le terme « clarify » (3,6) ou « referring » (3,5) comme dans « I'm referring to (…) » (message *t1_cndtf0x*). Au contraire d'un changement d'avis, les messages de classe négative peuvent dénoter la réaffirmation d'un propos avec le terme « already » (3,6) comme dans « I already said in different comments that (…) » (message *t1_cninoeb*).

Le descripteur POS fondé sur la fréquence des catégories morphosyntaxiques dans les messages donne 60,97%. Il utilise des traits de surface qui nécessitent tout de même une analyse linguistique plus approfondie que le sac de mots qui est remarquablement meilleur (cf. *Tableau 8*). Les traits discriminants pour la classe positive sont majoritairement les pronoms (3,0) pouvant dénoter la subjectivité d'un changement d'avis tandis que la classe négative est forte en coordonnants (0,8) et interjections (1,0) qui sont caractéristiques des débats.

Nous avons aussi essayé un trait un peu plus fondé sur la sémantique des messages en émettant l'hypothèse que lorsqu'ils expriment un changement d'avis, les auteurs font une sorte de bilan avec un retour sur le passé. La seule détection de toute forme du passé dans les messages donne un score de 57,15% (*Tableau 8*), ce qui semble valider notre hypothèse.

Nous avons utilisé des traits sur le style de l'expression dans les messages qui pourraient mettre en évidence un changement d'avis de leur auteur. Puisque le changement d'avis est lié à la psychologie, nous avons utilisé deux ressources lexicales construites empiriquement par des psychologues et utilisées aussi par (Tan et al., 2016) notamment pour détecter la malléabilité d'un auteur initial d'après son message introductif. Chaque lemme considéré reçoit un score selon qu'il évoque la gaieté, le contrôle et la passion dans (Warriner et al., 2013) et la factualité dans (Brysbaert et al., 2014). Nous calculons un score pour chaque message et pour chaque trait en faisant une moyenne sur les lemmes en présence. Nous ajoutons l'emploi des pronoms personnels de première personne pour l'aspect subjectif de l'expression d'un changement d'avis. Ce descripteur particulièrement complexe donne un score à 64,70%, ce qui est notablement moins bon que le sac de mots (cf. *Tableau 8*). Pour la classe positive, il y a un emploi caractéristique de la première personne du singulier avec le plus grand coefficient du modèle à 7,3, ce qui peut se rapporter à l'autocritique exercée par l'auteur qui explique son changement d'avis. Il y a aussi la présence particulière d'un lexique de contrôle avec un coefficient à 6,1 dénotant l'aspect sage et mesuré du bilan que constitue l'expression d'un changement d'avis. La classe négative est quant à elle particulièrement factuelle (2,3), ce que nous pouvons comprendre comme le soutien à une argumentation, et passionnée (2,6) puisque ce sont des messages qui sont dans la dynamique du débat.

4 Conclusion et Perspectives

La partie sur l'analyse de la structure montre une problématique réelle sur la faible connectivité d'utilisateurs pris pourtant dans un même domaine. Nous devons essayer d'augmenter la densité du graphe que nous construisons afin que les mesures de centralité puissent gagner en pertinence. Nous pourrions essayer de « résumer » le graphe obtenu en supprimant les nœuds à faible degré qui n'apportent pas vraiment d'information ou partir d'une communauté identifiée en amont pour travailler sur son réseau.

Nous allons essayer d'affiner la détection du changement d'avis en combinant les critères les plus pertinents. Nous tenterons ensuite d'identifier linguistiquement la source d'un changement d'avis. Nous pourrons utiliser un critère de similarité textuelle en partant de l'hypothèse que le message qui exprime le changement d'avis reprend les éléments déterminants du message qui a créé ce changement d'état.

En lien avec le changement d'avis comme un effet de l'influence, nous allons étudier ce qui provoque le changement d'avis, l'argumentation en partant des théories sur l'argumentation. Nous allons nous intéresser à la structure d'un argument pour voir s'il y a une composition « gagnante ».

Références

AMIGÓ, E., CARRILLO-DE-ALBORNOZ, J., CHUGUR, I. (2014). Overview of replab 2014: author profiling and reputation dimensions for online reputation management. Actes de *International Conference of the Cross-Language Evaluation Forum for European Languages*, 307-322

BIFET A., HOLMES G., PFAHRINGER B. (2011). Detecting sentiment change in Twitter streaming data. Actes de *2nd Workshop on Applications of Pattern Analysis* 17, 5-11

BIRAN O., ROSENTHAL S., ANDREAS J., MCKEOWN K., RAMBOW O. (2012). Detecting influencers in written online conversations. Actes de *Second Workshop on Language in Social Media*, 37-45

BRYSBAERT M., WARRINER A. B., KUPERMAN V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior research methods* 46, 904-911

DANLOS L. (2011). Analyse discursive et informations de factivité. Actes de *TALN 2011*

DAVE K., BHATT R., VARMA V. (2011). Identifying influencers in social networks. Actes de *5th International Conference on Weblogs and Social Media*, 1-9

FREEMAN, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks* 1-3, 215-239

GIONIS, A., TERZI, E., et TSAPARAS, P. (2013). Opinion maximization in social networks. Actes de *2013 SIAM International Conference on Data Mining*, 387-395

HAMZEHEI, A., JIANG, S., KOUTRA, D., WONG, R., CHEN, F. (2017). Topic-based Social Influence Measurement for Social Networks. *Australasian Journal of Information Systems*, 21

JABEUR, L. B., TAMINE L., BOUGHANEM M. (2012). Active microbloggers: Identifying influencers, leaders and discussers in microblogging networks. *International Symposium on String Processing and Information Retrieval*, 111-117

KHADANGI E., BAGHERI, A. (2017). Presenting novel application-based centrality measures for finding important users based on their activities and social behavior. *Computers in Human Behavior*

KLEINBERG J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 604-632

MARIANI J., PAROUBEK, P., FRANCOPOULO, G., HAMON O. (2014). Rediscovering 15 years of discoveries in language resources and evaluation: The LREC anthology analysis. Actes de *9th International Conference on Language Resources and Evaluation*

PAGE, L., BRIN, S., MOTWANI, R., WINOGRAD T. (1999). The PageRank citation ranking: Bringing order to the web. *Stanford InfoLab*

PAK A., PAROUBEK P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. Actes de *LREC* 10

QUERCIA D., ELLIS, J., CAPRA, L., CROWCROFT J. (2011). In the mood for being influential on twitter. Actes de *IEEE Third Inernational Conference on Social Computing (SocialCom)*, 307-314

SHEIKHAHMADI A., NEMATBAKHSH, M. A.,ZAREIE, A (2017). Identification of influential users by neighbors in online social networks. *Physica A: Statistical Mechanics and its Applications*

TAN, C., NICULAE, V., DANESCU-NICULESCU-MIZIL C., LEE L. (2016). Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. Actes de *25th international conference on world wide web*, 613-624

WARRINER A. B., KUPERMAN V., BRYSBAERT M. (2013). Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods* 45, 1191-1207

Détection d'influenceurs dans des médias sociaux

Kévin Deturck[1, 2]
(1) ERTIM, 2 rue de Lille, 75007 Paris, France
(2) Viseo, 4 avenue Doyen louis Weil, 38000 Grenoble, France
kevin.deturck@viseo.com

RÉSUMÉ

Les influenceurs ont la capacité d'avoir un impact sur d'autres individus lorsqu'ils interagissent avec eux. Détecter les influenceurs permet d'identifier les quelques individus à cibler pour toucher largement un réseau. Il est possible d'analyser les interactions dans un média social du point de vue de leur structure ou de leur contenu. Dans nos travaux de thèse, nous abordons ces deux aspects. Nous présentons d'abord une évaluation de différentes mesures de centralité sur la structure d'interactions extraites de Twitter puis nous analysons l'impact de la taille du graphe de suivi sur la performance de mesures de centralité. Nous abordons l'aspect linguistique pour identifier le changement d'avis comme un effet de l'influence depuis les messages d'un forum.

ABSTRACT

Influencer detection in social medias

There is an increasing interest in the detection of influencers on social medias. Different features may be used: the text of the messages and the structure of the network. The initial PhD works presented in this paper explore these two aspects. We evaluate the effectiveness of centrality measures from the state of the art in detecting Twitter influencers building graphs from interactions between Twitter users. In a second experimentation, we analyze the impact of the network size on the selection of the most appropriate centrality algorithms. We segment Twitter users according to the number of their followers and build their respective underlying "following graph". We then run the selected algorithms on the different graphs and evaluate their performance throughout the different graph sizes. We finally present a current work on linguistic features to detect opinion change as an influence effect.

MOTS-CLÉS : influence, média social, réseau, centralité, opinion

KEYWORDS: influence, social media, network, centrality, opinion

1 Introduction

Nos travaux de thèse visent à détecter les individus ayant une influence dans des médias sociaux. Nous entendons par "influence" le pouvoir que possède un individu qui parvient à mobiliser d'autres individus en faveur d'une action ou d'une opinion.

Détecter automatiquement les influenceurs dans des médias sociaux peut être vu comme une tâche pour un système de recherche d'information destiné à un utilisateur qui aurait besoin de connaitre les personnes meneuses et leurs propos dans un certain domaine d'activité. Les influenceurs constituent

aussi des points d'entrée efficaces pour la diffusion ciblée d'une information lors de campagnes de santé publique, pour la promotion d'un produit ou encore la gestion de réputation en ligne.

La pluralité des médias sociaux et la variété des informations qu'ils contiennent constituent autant de facettes possibles pour caractériser l'influence ouvrant ainsi de larges perspectives à l'élaboration d'un cadre formel permettant de la détecter. Il s'agit aussi d'une complexité supplémentaire pour modéliser l'information utile à la détection des influenceurs. Nous aurons à distinguer et à recouper la nature des informations disponibles à travers les différents médias sociaux pour les intégrer dans notre modèle. Par exemple, comparer la portée des interactions "Favori" dans Twitter et "J'aime" dans Facebook portant toutes les deux sur un contenu. Aussi, les textes que nous étudierons à travers les différents médias sociaux auront des formats tout à fait différents qui constitueront un corpus particulièrement hétérogène. Par exemple, nous ne pourrons pas aborder un texte de forum avec les mêmes modalités que celles utilisées pour un Tweet.

Les interactions entre les individus sont à la base de toute influence. Dans un média social, ces interactions peuvent être analysées d'après leur structure ou leur contenu. La structure peut porter une signification variée qui dépend à la fois des natures d'interaction considérées et de l'interprétation que nous leur prêtons. Nous nous situons dans le domaine du Traitement Automatique des Langues et nous focalisons donc nos travaux sur le contenu textuel des médias sociaux même si d'autres types de contenu comme l'image peuvent être utilisés par les influenceurs.

Identifier structurellement les influenceurs dans un groupe d'individus revient à trouver les clés de voûte de leurs interactions. Pour ce faire, nous devons trouver une représentation des interactions qui tienne compte de leurs natures variées. Par exemple, nous devons choisir comment inclure dans une même structure les interactions Twitter d'abonnement et de Retweet afin qu'elles soient les plus significatives possibles.

Nous avons à déterminer les phénomènes linguistiques qui permettent, du point de vue des influenceurs, de mettre en œuvre leur pouvoir et qui, du point de vue des individus cibles, font état de leurs réactions face aux influenceurs. Le principe est alors d'extraire des régularités expliquant sous différents types éventuels le phénomène d'influence par la langue écrite.

Les messages que nous devons analyser contiennent une expression informelle, un défi intéressant pour l'analyse linguistique parce que hors d'une grammaire standard de la langue. Pour l'analyse automatique, nous devrons mettre en œuvre un processus important de prétraitement permettant de ramener les textes des médias sociaux à une langue standard qui pourra être traitée par des outils de Traitement Automatique de la Langue. Le caractère elliptique des messages, en particulier dans les réseaux sociaux, peut empêcher l'identification de phénomènes linguistiques tels qu'ils sont traditionnellement décrits. Cela requiert une nouvelle modélisation des phénomènes qu'on veut repérer pour qu'ils soient applicables aux modes d'expression récents que nous souhaitons analyser. Prenons l'exemple d'un influenceur qui argumente pour changer l'opinion de quelqu'un. Pour caractériser son argumentation à l'intérieur d'un Tweet et la reconnaitre dans d'autres messages, nous ne pourrons certainement pas utiliser exactement les modèles traditionnels de l'argumentation.

2 État de l'art

2.1 Sur la détection structurelle des influenceurs

La centralité permet de mesurer l'importance structurelle d'un nœud dans un graphe. Nous pouvons analyser un média social par la structure de ses interactions représentées comme les arcs d'un graphe. Dans un réseau, l'influenceur est un utilisateur qui polarise les interactions, c'est donc un nœud central du graphe correspondant.

Le positionnement d'un individu dans un réseau est déterminé par ses connexions avec d'autres utilisateurs. Ces connexions sont données par les interactions de tous types entre les individus d'un média social. C'est plus particulièrement les réseaux sociaux qui sont à l'étude ici puisque la nature même du réseau rend essentielle l'analyse de ses liens. Un influenceur est alors identifié par un positionnement central à l'intérieur du réseau. Différentes mesures de centralité ont été proposées donnant lieu à des interprétations d'influence différentes. (Mariani et al., 2014) analysent les réseaux de citations pour mesurer l'influence scientifique. Ils appliquent une mesure de centralité qui prend en compte la proximité d'un utilisateur par rapport aux autres dans le graphe selon des liens que sont les citations des publications et les collaborations. La valeur de centralité pour chaque utilisateur indique son degré d'influence. (Sheikhahmadi et al., 2017) mettent en avant l'importance de la communauté dans les interactions des utilisateurs d'un réseau social. Les auteurs utilisent la tendance des utilisateurs à communiquer à l'intérieur d'un groupe constitué par exemple autour d'une thématique. Ils affirment donc que les influenceurs doivent être identifiés pour chaque communauté et divisent donc le graphe par détection de communauté avant d'utiliser la quantité d'interactions pour pondérer l'influence de chaque utilisateur dans sa communauté. (Khadangi, Bagheri, 2017) distinguent les interactions marquant une affinité entre les utilisateurs de celles qui reflètent l'activité propre de d'un utilisateur comme les « J'aime » sur Facebook, ces dernières étant moins étudiées pour l'influence.

Puisque les influenceurs ont la capacité de mobiliser d'autres individus pour des actions ou des opinions, nous pouvons aussi les identifier en analysant les dynamiques dans la structure des interactions. Un influenceur est alors un individu qui parvient à propager une attitude le plus rapidement et le plus longuement à travers un réseau. (Dave et al., 2011) cherchent à prédire quelle sera la dynamique de propagation d'une action à partir d'un certain utilisateur pour prédire son influence. (Jabeur et al., 2012) voient plus généralement les influenceurs comme ayant une capacité à propager une information. Les auteurs se concentrent ainsi, dans Twitter, sur les Retweets pour construire un graphe d'utilisateurs qu'ils analysent à la manière de Page Rank en considérant l'importance des utilisateurs Retweetant pour calculer l'influence de l'utilisateur Retweeté. (Gionis et al., 2013) s'intéressent à l'opinion des utilisateurs pour prédire ceux dont elle va le mieux se maximiser à travers un réseau d'après la constitution de leur voisinage respectif. Ces utilisateurs sont les clés du problème de maximisation d'influence à travers un réseau social. Le principe est de trouver l'ensemble d'utilisateurs qui permettra de diffuser une information le plus rapidement possible dans un réseau.

2.2 Sur la détection linguistique des influenceurs

Les approches qui analysent les textes issus de médias sociaux pour détecter les influenceurs, comme (Biran et al., 2012), essayent d'identifier les marqueurs d'un discours influent. L'influenceur

s'exprime souvent pour soutenir une revendication à propos d'un certain sujet. Cette revendication contient un point de vue, elle se caractérise donc essentiellement par une énonciation subjective. À ce propos, (Pak, Paroubek, 2010) cherchent à repérer automatiquement le point de vue adopté dans des Tweets par apprentissage automatique sur les catégories morphosyntaxiques en présence. Les auteurs mettent en avant l'utilisation d'adjectifs superlatifs pour renforcer l'expression des émotions et la forte présence de pronoms personnels indiquant une personnalisation du discours comme des traits particulièrement discriminants pour le discours subjectif. En général, l'influenceur essaye de faire adhérer d'autres personnes à son discours en utilisant deux grands procédés argumentatifs : persuader et convaincre.

La persuasion se caractérise par une implication fortement personnelle qui s'exprime avec des émotions et des opinions. La richesse des émotions exprimées peut, dans le cas de la persuasion, compenser l'absence de raisonnement. Persuader fait appel à une argumentation intuitive. La répétition d'un même propos entre ainsi dans le cadre de la persuasion.

Par contraste avec la persuasion, convaincre nécessite le développement d'une argumentation plus rationnelle pour servir la revendication exprimée. Il s'agit alors d'un discours moins direct fondé sur le raisonnement. (Rosenthal, 2014)

Détecter un texte argumentatif revient à faire de l'analyse du discours. L'argumentation est décrite en certaines relations rhétoriques présentées dans (Biran & Rambow, 2011) qui lient une partie revendicative à une partie justificative. Chacune de ces parties constitue une unité de discours à identifier. Une fois les unités de discours élémentaires identifiées, il est question de trouver les relations rhétoriques qu'il y a entre elles (Danlos, 2011). Ces relations permettent de structurer l'argumentation.

(Quercia et al., 2011) s'intéressent à l'usage de la langue chez les utilisateurs de Twitter dans une approche statistique pour identifier les influenceurs. Ils mettent en avant l'usage d'un champ lexical de l'émotion pour instaurer une intimité avec les lecteurs et ainsi faciliter l'impact du message à transmettre. Cette intimité est complétée par un usage de la troisième personne pour donner aux lecteurs l'impression d'appartenance à une même communauté qui se démarque des autres.

Pour repérer l'évolution d'une opinion sur Twitter, (Bifet et al., 2011) analysent la spécificité d'utilisation des termes en présence pour chaque Tweet d'un utilisateur à travers des fenêtres temporelles. Ils considèrent la polarité positive ou négative du lexique pour attribuer une valeur à l'opinion et ils utilisent les Hashtags pour créer des ensembles de Tweets portant sur une même thématique.

L'analyse linguistique peut aussi servir à orienter la détection d'influenceurs vers une thématique, nous avons mentionné précédemment l'importance de la communauté, notamment thématique, pour l'exercice d'une influence. (Hamzehei et al., 2017) analysent l'influence des utilisateurs de réseaux sociaux à l'aune des topiques détectés dans leurs messages.

3 Expérimentations réalisées

Nous avons, en section 1, défini l'influence comme un pouvoir, ce qui nous amène à caractériser les influenceurs selon deux aspects : les ressources qui leur permettent d'influencer et les effets de leur influence du point de vue des individus cibles. Nous allons rechercher les ressources comme les

effets des influenceurs aussi bien dans le contenu des messages échangés entre les influenceurs et leur audience que sur des caractéristiques topologiques concernant la structure et la nature de leurs interactions. Nos expérimentations comportent ainsi deux pans avec la relations d'un individu Nous en déduisons de nouvelles pistes afin de créer un système hybride entre les aspects de structure et de contenu souvent étudiés indépendamment.

3.1 Comparaison de mesures de centralité pour la détection d'influenceurs Twitter

Le but de cette expérimentation est d'évaluer des algorithmes mesurant chacun une certaine centralité. Nous pouvons distinguer deux types de centralité. La centralité locale qui prend en compte uniquement le voisinage d'un nœud pour lui attribuer une valeur de centralité. La centralité globale qui regarde le nœud par rapport à l'ensemble du graphe afin d'évaluer sa centralité. Pour chaque mesure de centralité, nous évaluons sa capacité à modéliser l'influence des individus représentés comme les nœuds d'un graphe qui représente la structure de leurs interactions.

Nous présentons six mesures de centralité qui nous permettent d'évaluer autant de types de centralité.

- **Degré entrant** : c'est une mesure de centralité locale puisqu'elle n'est basée que sur le calcul du nombre de liens entrants pour un nœud donné. En termes d'influence, c'est une polarisation directe envers un utilisateur (Freeman, 1978)

- **Intermédiarité** et **Proximité** : il n'y a pas de résultat pour Intermédiarité et Proximité parce qu'ils ont besoin de calculer les chemins les plus courts entre tous les nœuds du graphe, ce qui nécessite un graphe connecté, ce n'est pas le cas des graphes que nous avons extraits (Freeman, 1978) Nous les mentionnons tout de même parce qu'elles sont des mesures importantes qui ont bien été prises en compte dans nos expérimentations.

- **Hits** : apporte une sémantique supplémentaire à la centralité globale avec une distinction mutuelle entre un score d'autorité et un score de relais. L'autorité est importante parce que particulièrement écoutée par des relais qui sont eux-mêmes importants parce qu'ils écoutent beaucoup d'autorités (Kleinberg, 1999)

- **Page Rank** : calcule l'importance d'un nœud dans un graphe d'après la valeur de chaque lien pointant vers lui et l'importance du nœud à l'origine, avec une initialisation uniforme des poids et par itérations successives sur tous les nœuds du graphe jusqu'à ce que le poids de chacun soit à l'équilibre, cet algorithme donne des valeurs de centralité globale. La valeur de chaque se dilue avec le nombre de liens sortants de son nœud d'origine. Est ajoutée une probabilité uniforme pour tous les nœuds du graphe de se départir de la structure du réseau pour « sauter » d'un nœud à l'autre (Page et al., 1999)

- **Leader Rank** : entend améliorer la modélisation de Page Rank pour les réseaux sociaux en affirmant que la probabilité de passage d'un nœud à un autre sans utiliser les arcs du graphe ne doit pas être uniforme mais inversement proportionnelle au nombre de liens sortants qui sont disponibles

Ces algorithmes attribuent à chacun des nœuds du graphe un score de centralité permettant d'établir un classement d'utilisateurs pour un réseau considéré. Pour les évaluer, nous devons mesurer la qualité du classement produit à partir de chaque algorithme en fonction d'une référence qui donne l'influence de chaque utilisateur. Nous avons ajouté une Baseline fondée sur un classement au hasard des utilisateurs pour mieux appréhender les résultats des algorithmes évalués.

Comme référence, nous avons choisi le jeu de données conçu pour la compétition RepLab 2014 (Amigó et al., 2014). Cette compétition proposait notamment une tâche consistant à classer des utilisateurs Twitter du plus influent au moins influent. Le jeu de données contient plus de 7000 comptes Twitter catégorisés selon qu'ils appartiennent au domaine de la banque, de l'automobile ou à des domaines différents des deux précédents.. Ils ont été annotés par les spécialistes de la réputation en ligne Llorente & Cuenca[1]. Cette annotation considère l'influence réelle (dans le monde) des utilisateurs, en indiquant s'ils sont influenceurs ou pas. Le jeu de données contient en moyenne 1/3 d'influenceurs. Nous avons construit un échantillon de 50 utilisateurs du domaine de la banque pour limiter la taille du graphe à construire et le temps d'extraction des informations depuis l'API Twitter[2], qui impose des limites à la quantité d'information extraite par intervalle de temps. Nous avons fait en sorte de conserver la proportion d'influenceurs originale (1/3) pour garder une certaine comparabilité avec les systèmes de la compétition.

À partir de cet ensemble d'utilisateurs initial, nous extrayons deux types d'interaction.

- Le **suivi** qui constitue une audience et donc un terreau d'utilisateurs pour l'exercice d'une influence

- Le **Retweet** ou la rediffusion d'un contenu constituant une réaction comme un effet d'influence

À partir des utilisateurs initiaux, nous extrayons ces deux types d'interaction et donc de nouveaux utilisateurs qui seront représentés dans un graphe pour chacun des deux types d'interaction.

Nous comparons la référence binaire aux classements issus des algorithmes avec la mesure Mean Average Precision (MAP) utilisée pour RepLab 2014. Elle est fondée sur l'intuition en Recherche d'Information que les résultats les plus pertinents, les influenceurs, doivent apparaitre au début. Nous avons calculé MAP par la formule qui suit :

$$MAP = \frac{1}{n} \sum_{i=1}^{N} p(i) R(i)$$

avec N le nombre total d'utilisateur, n le nombre d'influenceurs correctement trouvés, $p(i)$ la précision au rang i (en ne considérant que les i premiers utilisateurs trouvés) et $R(i)$ est à 1 si l'utilisateur au rang i est un influenceur sinon à 0.

[1] www.llorenteycuenca.com/en/
[2] www.developer.twitter.com

Sur un graphe de suivi

Nombre de nœuds	5,067,480
Nombre d'arcs	5,149,491
Densité	10^{-7}

Tableau 1 : caractéristiques du graphe de suivi construit

Algorithme	Baseline	Degré entrant	Page Rank	Leader Rank	Hits
MAP (%)	**38,67**	**43,49**	**44,28**	**44,53**	**51,68**

Tableau 2 : résultats des mesures de centralité sur le graphe de suivi

- On observe *Tableau 1* une faible densité du graphe, les utilisateurs du corpus initial se suivent peu, ce qui explique pourquoi Page Rank et Leader Rank, mesures de centralité globales, donnent *Tableau 2* des résultats similaires au degré entrant, locale

- Hits se démarque en distinguant les influenceurs comme étant des « autorités », ce qui permet de donner comme première caractéristique des influenceurs d'être suivis par des « relais » tels que nous les avons précédemment présentés.

Sur un graphe de Retweet

Nombre de nœuds	2099
Nombre d'arcs	2051
Densité	10^{-4}

Tableau 3 : caractéristiques du graphe de Retweet construit

Algorithme	Baseline	Degré entrant	Page Rank	Leader Rank	Hits
MAP (%)	**38,67**	**40,91**	**40,91**	**40,91**	**40,91**

Tableau 4 : résultats des mesures de centralité sur le graphe de Retweet

- Même problème de connectivité que sur le graphe de suivi pour expliquer l'absence de résultat d'Intermédiarité et de Proximité

- À nouveau, la faible densité du graphe, signifiant ici que les utilisateurs se retweetent peu, empêche les mesures de centralité globales d'apporter de meilleurs résultats par rapport au degré entrant, local, qui ne prend en compte que le voisinage direct

- Le fait que Hits ne se distingue pas sur le graphe de Retweet peut indiquer que l'information de suivi est plus significative pour détecter les influenceurs.

Algorithme	UTDBRG	Lys	LIA	UAMCLYR	ORM_UNED
MAP	**0,41**	**0,52**	**0,45**	**0,49**	**0,32**

Tableau 5 : Résultats des système de RepLab sur la banque

Pour information, nous ajoutons *Tableau 5* les résultats des systèmes ayant participé à la compétition RepLab et diffusés dans (Amigó et al., 2014). Nous avons pris le meilleur essai pour chaque système sur l'ensemble du corpus et sur le même domaine que notre étude. Les meilleurs systèmes ont surtout utilisé des métadonnées des profils comme le texte de présentation, le nombre de suiveurs (Lys, UAMCLYR), la présence d'une image de profil et le statut de vérification du compte (Lys).

3.2 Étude de l'impact du nombre de suiveurs pour distinguer des influenceurs Twitter par des mesures de centralité

Nous nous concentrons pour cette expérimentation sur l'information de suivi qui nous a semblé plus significative. Nous restreignons notre sélection de mesures de centralité à Page Rank, Leader Rank et Hits qui n'ont pas besoin d'un graphe connexe pour calculer un résultat. Ces trois algorithmes opèrent itérativement pour calculer un score approximé tolérant pour la convergence une certaine variation entre deux itérations. Nous cherchons à la fois à analyser le comportement des mesures de centralité en faisant évoluer la taille du graphe de suivi et aussi à déterminer dans quelle mesure le nombre de suiveurs est significatif en l'utilisant dans des intervalles de valeur différents.

Nous utilisons le même corpus que précédemment en créant cette fois-ci des échantillons de 50 utilisateurs suivant les intervalles de nombre de suiveurs : [1000-5000], [5000-10,000], [10,000-50,000], [50,000-100,000], [100,000-500,000]. Nous faisons en sorte que chaque échantillon comprenne la même proportion d'influenceurs : 1/3 (proportion d'influenceurs du corpus d'origine). Nous construisons un graphe de suivi pour chacun des segments ce qui donne cinq graphes de suivi aux tailles différentes sur lesquelles nous appliquons les mesures de centralité précédemment sélectionnées.

Segments/Caractéristiques	**[1000-5000]**	**[5000-10,000]**	**[10,000-50,000]**	**[50,000-100,000]**	**[100,000-500,000]**
Nombre de nœuds	122,060	361,422	1,000,180	3,156,671	9,708,482
Nombre d'arcs	124,747	372,939	1,034,110	3,322,007	10,521,923
Densité	10^{-6}	10^{-6}	10^{-6}	10^{-7}	10^{-7}

Tableau 6 : Caractéristiques des graphes de suivi selon les segments

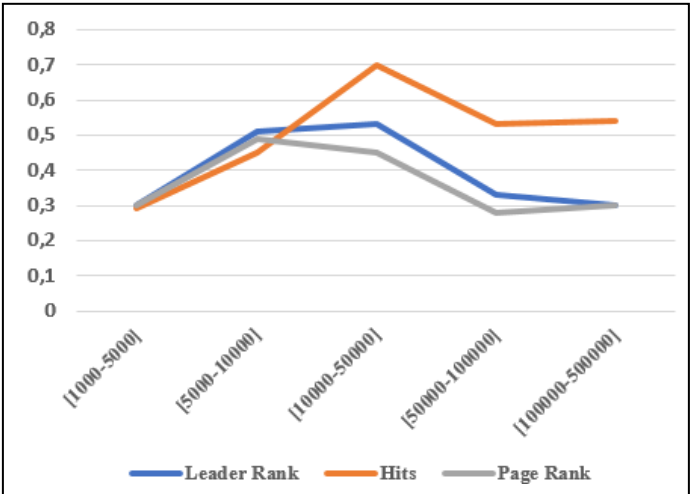


Figure 1 : Performances MAP selon les segments

La taille des segments augmentant, l'écart possible en nombre de suiveurs entre les utilisateurs augmente aussi, ce qui accroît potentiellement le pouvoir discriminant de ce critère pour distinguer les influenceurs. Le fait que la performance des trois algorithmes augmente jusqu'à un certain point avec la taille des segments montre que l'information de suivi est effectivement significative pour la détection des influenceurs. Nous expliquons globalement la baisse des performances en disant que le taille du graphe devient trop importante (*Tableau 3*, facteur 3 d'agrandissement du graphe entre deux segments) pour que l'approximation calculée par les trois algorithmes suffise à représenter correctement les forces en présence dans tout le graphe.

Nous constatons *Figure* 1 que deux algorithmes sont plus stables : Page Rank et Leader Rank. Leur comportement similaire peut s'expliquer par le fait que le second est une modification du premier. Aussi, ces deux algorithmes ont la particularité d'autoriser le fait de ne pas suivre la structure du graphe pour passer d'un nœud à l'autre, cela explique pourquoi ils sont moins sensibles aux variations de taille du graphe.

Pour Hits, la dynamique est plus forte (50% de progression relative entre [1000-5000] et [10,000-50,000] et 15% de baisse absolue entre [10,000-50,000] et [100,000-500,000]). Contrairement aux deux algorithmes précédents, Hits suit strictement la structure du graphe : pas de « saut » d'un nœud à l'autre ni de dilution de la valeur des liens. Il est donc plus sensible à la modification de la taille du graphe. Le fait que l'augmentation de marge en nombre de suiveurs entre les utilisateurs aide l'algorithme qui se base le plus sur la structure montre aussi la significativité de l'information de suivi.

La modélisation « nœud de terre » dans Leader Rank tient à mieux modéliser le comportement des utilisateurs d'un réseau social lorsque le nombre de liens sortants augmente et donc lorsque la taille du graphe augmente. Or, nous pouvons constater *Figure 1* que Leader Rank se détache positivement de Page Rank lorsque la taille des segments croît. C'est un signal que la modélisation de Leader Rank est effectivement meilleure que celle de Page Rank.

3.3 Détection linguistique du changement d'avis comme un effet de l'influence dans un corpus "Change My View[3]"

Notre parti pris est de détecter les influenceurs par leur influence effective là où des travaux précédents émettent des hypothèses sur l'influençabilité de critères linguistiques. Ainsi, nous voyons la détection du changement d'avis comme une première étape à la détection d'influenceurs en tant qu'un effet d'influence. Il s'agirait pour la suite de repérer la source du changement d'avis pour identifier l'influenceur. L'objectif est d'obtenir des influenceurs réels plutôt que des potentiels influenceurs.

Corpus/Caractéristiques	#fils de discussion Pour l'entrainement	#messages Pour l'entrainement	#fils de discussion Pour l'évaluation	#messages Pour l'évaluation
Initial	18,363	1,114,533	2,263	145,733
Pré-filtré	10,743	128,901	1,529	20,883
Final	3,191	42,776	672	9463

Tableau 7 : Statistiques générales des corpus jusqu'au filtrage final

Nous avions besoin d'une ressource contenant des manifestations textuelles de changement d'avis. Nous avons choisi le forum en anglais « Change My View », précédemment étudié pour des tâches connexes par (Tan et al., 2016). Le principe de ce forum est que l'auteur initial d'un fil de discussion expose son point de vue sur une thématique puis il demande aux lecteurs de le faire changer d'avis. Les autres participants au fil de discussion vont alors faire en sorte de développer une argumentation qui contredise assez bien le point de vue de l'auteur initial pour modifier son point de vue. Lorsque l'auteur initial reconnait qu'un message a eu l'effet escompté, il lui attribue un « delta », ce qui revient à une annotation « ad hoc » des messages gagnants. C'est cette annotation que nous utiliserons comme référence. L'auteur initial doit en plus justifier cette récompense dans un message qui explicite son changement d'avis.

Nous utilisons le corpus extrait par (Tan et al., 2016) qui essayent notamment de prédire les arguments qui vont créer un changement d'avis. Toujours dans l'optique de détecter un influenceur effectif, nous adoptons une approche inverse puisque nous partons des réponses à ces arguments pour détecter ceux qui ont eu un impact. Les auteurs ont extrait plus de 20,000 fils de discussion depuis la création du forum en 2013 jusqu'en 2015 (les statistiques du corpus initial sont dans le *Tableau 6*). Cela donne plus de 1,000,000 de messages pour environ 80,000 participants uniques, donc beaucoup d'habitués. Le corpus est déjà divisé entre une partie pour l'entrainement (90%) et une partie pour l'évaluation (10%). Nous partons du corpus que les auteurs ont filtré pour leur tâche sur la résistance à la persuasion (statistiques générales du corpus pré-filtré en *Tableau 6*). Ils posent des contraintes de pertinence sur une participation minimale pour l'auteur initial et pour les autres à l'intérieur d'un fil de discussion. Puisque nous cherchons à détecter le changement d'avis pour les auteurs initiaux, nous analysons leurs messages seulement dans les fils de discussion où ils changent d'avis au moins une fois (les statistiques générales du corpus final en *Tableau 6*). Les discussions qui contiennent au moins un changement d'avis représentent 30% du corpus pré-filtré par les auteurs (*Tableau 6*). Dans le corpus final, le taux d'exemples positifs (messages d'un auteur initial exprimant un changement d'avis) est de 10%, ce qui réduit légèrement le déséquilibre entre les

[3] www.reddit.com/r/changemyview/

classes (2% initialement). Tous les messages sont analysés sans distinction d'auteur. Nous avons supprimé des messages les marques liées à l'attribution d'une récompense comme « delta » pour ne pas biaiser la classification.

Nous pouvons voir ce travail comme une tâche de classification binaire puisque pour chaque message, nous devons dire s'il exprime un changement d'avis ou pas. Nous utilisons un classifieur par régression logistique qui convient particulièrement à ce genre de classification. Le corpus original comme notre échantillon d'entrainement contiennent seulement 2% de messages exprimant un changement d'avis. Cela constitue un biais pour l'apprentissage du classifieur qui pourrait simplement annoter les messages avec la classe majoritaire (pas de changement d'avis) afin d'obtenir un résultat correct. Pour contrebalancer ce biais, nous mesurons la performance de notre classifieur en utilisant la mesure Area Under ROC Curve qui prend en compte le taux de vrais positifs par rapport au taux de faux positifs.

Dans cette expérimentation, toujours en cours, nous avons commencé par déterminer les descripteurs les plus pertinents en les utilisant séparément. Dans la poursuite de nos travaux, nous travaillerons sur la combinaison de ces traits pour obtenir le meilleur résultat possible.

Descripteur	Nombre de mots (référentiel)	Sac de mots	POS	Style	Passé
Score AUC (%)	**51,38**	**82,11**	**60,97**	**64,70**	**57,15**

Tableau 8 : Résultats de descripteurs pour la détection de changement d'avis

Le trait le plus simple, qui constitue le référentiel de notre évaluation, consiste à utiliser le nombre de mots du message. Nous obtenons un résultat assez neutre pour un référentiel à 51,38% (cf. *Tableau 8*).

Nous nous sommes demandé s'il y avait un emploi de termes particulier pour les messages exprimant un changement d'avis. Nous utilisons une représentation en sac de mots des tokens présents dans chaque message. Nous obtenons le meilleur résultat avec 82,11% pour ce trait seul (*Tableau 8*). Nous avons analysé les termes les plus discriminants pour donner du sens à ce résultat chiffré. Pour la classe positive (changement d'avis), le terme au poids le plus important (10,4) est « convinced », qui est central pour exprimer l'impact d'un argument gagnant comme dans « This one finally convinced me » (message *t1_cna7wg7*). La classe positive contient des termes de concession comme « concede », « still » qui marquent un tournant. Nous observons aussi un poids très important (8,1) pour « hadn » qui semble marquer une remise en cause du point de vue passé comme dans « I hadn't considered the obvious fact that (…) » (message *t1_cnbkumq*). Toujours dans ce modèle avec un terme au passé à polarité négative, nous observons l'émergence du terme « forgot » (6,5). Enfin, il y a un ensemble de termes ayant trait à la clairvoyance tels que « guesss » ou « realize » (7), qui dénotent la compréhension d'une nouvelle vision des choses comme dans « I think this post really helped me realize (…) » (message *t1_cni1gsy*). Pour la classe négative (messages sans expression d'un changement d'avis), le terme le plus fort (4,7) est « talking », qui peut à la fois montrer que la discussion n'est pas résolue et qu'il y a un malentendu comme dans « I am talking about (…) » (message *t1_cngt9pv*). Ce malentendu transparait aussi dans les messages de la classe négative avec le terme « clarify » (3,6) ou « referring » (3,5) comme dans « I'm referring to (…) » (message *t1_cndtf0x*). Au contraire d'un changement d'avis, les messages de classe négative peuvent dénoter la réaffirmation d'un propos avec le terme « already » (3,6) comme dans « I already said in different comments that (…) » (message *t1_cninoeb*).

Le descripteur POS fondé sur la fréquence des catégories morphosyntaxiques dans les messages donne 60,97%. Il utilise des traits de surface qui nécessitent tout de même une analyse linguistique plus approfondie que le sac de mots qui est remarquablement meilleur (cf. *Tableau 8*). Les traits discriminants pour la classe positive sont majoritairement les pronoms (3,0) pouvant dénoter la subjectivité d'un changement d'avis tandis que la classe négative est forte en coordonnants (0,8) et interjections (1,0) qui sont caractéristiques des débats.

Nous avons aussi essayé un trait un peu plus fondé sur la sémantique des messages en émettant l'hypothèse que lorsqu'ils expriment un changement d'avis, les auteurs font une sorte de bilan avec un retour sur le passé. La seule détection de toute forme du passé dans les messages donne un score de 57,15% (*Tableau 8*), ce qui semble valider notre hypothèse.

Nous avons utilisé des traits sur le style de l'expression dans les messages qui pourraient mettre en évidence un changement d'avis de leur auteur. Puisque le changement d'avis est lié à la psychologie, nous avons utilisé deux ressources lexicales construites empiriquement par des psychologues et utilisées aussi par (Tan et al., 2016) notamment pour détecter la malléabilité d'un auteur initial d'après son message introductif. Chaque lemme considéré reçoit un score selon qu'il évoque la gaieté, le contrôle et la passion dans (Warriner et al., 2013) et la factualité dans (Brysbaert et al., 2014). Nous calculons un score pour chaque message et pour chaque trait en faisant une moyenne sur les lemmes en présence. Nous ajoutons l'emploi des pronoms personnels de première personne pour l'aspect subjectif de l'expression d'un changement d'avis. Ce descripteur particulièrement complexe donne un score à 64,70%, ce qui est notablement moins bon que le sac de mots (cf. *Tableau 8*). Pour la classe positive, il y a un emploi caractéristique de la première personne du singulier avec le plus grand coefficient du modèle à 7,3, ce qui peut se rapporter à l'autocritique exercée par l'auteur qui explique son changement d'avis. Il y a aussi la présence particulière d'un lexique de contrôle avec un coefficient à 6,1 dénotant l'aspect sage et mesuré du bilan que constitue l'expression d'un changement d'avis. La classe négative est quant à elle particulièrement factuelle (2,3), ce que nous pouvons comprendre comme le soutien à une argumentation, et passionnée (2,6) puisque ce sont des messages qui sont dans la dynamique du débat.

4 Conclusion et Perspectives

La partie sur l'analyse de la structure montre une problématique réelle sur la faible connectivité d'utilisateurs pris pourtant dans un même domaine. Nous devons essayer d'augmenter la densité du graphe que nous construisons afin que les mesures de centralité puissent gagner en pertinence. Nous pourrions essayer de « résumer » le graphe obtenu en supprimant les nœuds à faible degré qui n'apportent pas vraiment d'information ou partir d'une communauté identifiée en amont pour travailler sur son réseau.

Nous allons essayer d'affiner la détection du changement d'avis en combinant les critères les plus pertinents. Nous tenterons ensuite d'identifier linguistiquement la source d'un changement d'avis. Nous pourrons utiliser un critère de similarité textuelle en partant de l'hypothèse que le message qui exprime le changement d'avis reprend les éléments déterminants du message qui a créé ce changement d'état.

En lien avec le changement d'avis comme un effet de l'influence, nous allons étudier ce qui provoque le changement d'avis, l'argumentation en partant des théories sur l'argumentation. Nous allons nous intéresser à la structure d'un argument pour voir s'il y a une composition « gagnante ».

Références

AMIGÓ, E., CARRILLO-DE-ALBORNOZ, J., CHUGUR, I. (2014). Overview of replab 2014: author profiling and reputation dimensions for online reputation management. Actes de *International Conference of the Cross-Language Evaluation Forum for European Languages*, 307-322

BIFET A., HOLMES G., PFAHRINGER B. (2011). Detecting sentiment change in Twitter streaming data. Actes de *2nd Workshop on Applications of Pattern Analysis* 17, 5-11

BIRAN O., ROSENTHAL S., ANDREAS J., MCKEOWN K., RAMBOW O. (2012). Detecting influencers in written online conversations. Actes de *Second Workshop on Language in Social Media*, 37-45

BRYSBAERT M., WARRINER A. B., KUPERMAN V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior research methods* 46, 904-911

DANLOS L. (2011). Analyse discursive et informations de factivité. Actes de *TALN 2011*

DAVE K., BHATT R., VARMA V. (2011). Identifying influencers in social networks. Actes de *5th International Conference on Weblogs and Social Media*, 1-9

FREEMAN, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks* 1-3, 215-239

GIONIS, A., TERZI, E., et TSAPARAS, P. (2013). Opinion maximization in social networks. Actes de *2013 SIAM International Conference on Data Mining*, 387-395

HAMZEHEI, A., JIANG, S., KOUTRA, D., WONG, R., CHEN, F. (2017). Topic-based Social Influence Measurement for Social Networks. *Australasian Journal of Information Systems*, 21

JABEUR, L. B., TAMINE L., BOUGHANEM M. (2012). Active microbloggers: Identifying influencers, leaders and discussers in microblogging networks. *International Symposium on String Processing and Information Retrieval*, 111-117

KHADANGI E., BAGHERI, A. (2017). Presenting novel application-based centrality measures for finding important users based on their activities and social behavior. *Computers in Human Behavior*

KLEINBERG J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 604-632

MARIANI J., PAROUBEK, P., FRANCOPOULO, G., HAMON O. (2014). Rediscovering 15 years of discoveries in language resources and evaluation: The LREC anthology analysis. Actes de *9th International Conference on Language Resources and Evaluation*

PAGE, L., BRIN, S., MOTWANI, R., WINOGRAD T. (1999). The PageRank citation ranking: Bringing order to the web. *Stanford InfoLab*

PAK A., PAROUBEK P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. Actes de *LREC* 10

QUERCIA D., ELLIS, J., CAPRA, L., CROWCROFT J. (2011). In the mood for being influential on twitter. Actes de *IEEE Third Inernational Conference on Social Computing (SocialCom)*, 307-314

SHEIKHAHMADI A., NEMATBAKHSH, M. A.,ZAREIE, A (2017). Identification of influential users by neighbors in online social networks. *Physica A: Statistical Mechanics and its Applications*

TAN, C., NICULAE, V., DANESCU-NICULESCU-MIZIL C., LEE L. (2016). Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. Actes de *25th international conference on world wide web*, 613-624

WARRINER A. B., KUPERMAN V., BRYSBAERT M. (2013). Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods* 45, 1191-1207

Classification par paires de mention pour la résolution des coréférences en français parlé interactif

Maëlle Brassier[1], Alexis Puret[2] , Augustin Voisin-Marras[2], Loïc Grobol[3]
(1) LIFAT, Université de Tours, 3 place Jean Jaurès, 41000 Blois, France
(2) LIFO, Université d'Orléans, 6 Rue Léonard de Vinci, 45067 Orléans, France
(3) LATTICE-CNRS, ENS, Université Paris 3, Université Sorbonne Paris Cité.

maelle.brassier@yahoo.fr, {alexis.puret,augustin.voisin-marras}@etu.univ-orleans.fr, loic.grobol@gmail.com

RÉSUMÉ

Cet article présente et analyse les premiers résultats obtenus par notre laboratoire pour la construction d'un modèle de résolution des coréférences en français à l'aide de techniques de classifications parmi lesquelles les arbres de décision et les séparateurs à vaste marge. Ce système a été entraîné sur le corpus ANCOR et s'inspire de travaux antérieurs réalisés au laboratoire LATTICE (système CROC). Nous présentons les expérimentations que nous avons menées pour améliorer le système en passant par des classifieurs spécifiques à chaque type de situation interactive, puis chaque type de relation de coréférence.

ABSTRACT

Mention-pair classification for corefence resolution on spontaneous spoken French.
This paper presents the first experiments conducted by our laboratory (LIFAT) on the question of the resolution of coreference on spontaneous spoken French. We have developed a mention-pair classifier, trained on the ANCOR French coreference corpus, which is based on various classification techniques among which support vector machines (SVM). The paper details several experimental studies that investigate several factors (classification model, interactivity degree, nature of the coreference…) that should affect the performances of the system.

MOTS-CLÉS : détection de coréférence, corpus, apprentissage automatique, classification
KEYWORDS: coreference resolution, corpus, machine learning, classification

1. Introduction

Depuis sa création, le TAL a engendré de nombreuses technologies qui ont permis le développement d'applications comme la traduction et la compréhension de textes pour lesquelles la question de la coréférence, est un enjeu essentiel. Afin de s'assurer de l'interprétation correcte des textes étudiés, il est en effet important de relier les expressions linguistiques aux entités du discours auxquelles elles réfèrent. Une simple erreur au cours de cette tâche peut entraîner un contresens radical d'une phrase. La détection de la coréférence, qui nous intéresse ici, consiste à relier les expressions

(appelées mentions) qui réfèrent la même entité du discours. Considérons l'énoncé (A) ci-dessous : *Le roi de la pop* et *Michael Jackson* font référence à la même entité du discours (en l'occurrence une personne) tout en ayant des dénominations différentes.

A. **Michael Jackson** est mort en 2009. **Le roi de la pop** aura influencé de nombreux artistes. (*coréférence indirecte*)

On distingue la coréférence de l'anaphore, qui décrit seulement le fait que l'interprétation d'une mention dépend de celle d'un autre élément linguistique, son antécédent. Les notions d'anaphore et de coréférence sont étroitement liées, mais sont néanmoins distinctes. En effet, une anaphore ne manifeste pas forcément une relation coréférente puisqu'elle peut reprendre une expression déjà introduite dans le discours sans pour autant désigner la même entité. Dans l'exemple (B), le groupe nominal *Sa toiture* est anaphorique de *La maison* bien que les deux mentions soient non-coréférentes.

B. **La maison** est délabrée. **Sa toiture** tombe en ruine. *(Anaphore nominale*)

À ce jour, peu de systèmes de résolution des coréférences adaptés à la langue française ont été mis en place, malgré l'existence d'un corpus français annoté qui permet de développer des systèmes par apprentissage automatique. Ce papier présente les premiers développements d'un nouveau système de résolution par apprentissage automatique s'inspirant des travaux de (Désoyer et al., 2015) sur le système CROC (Adèle Désoyer et al., 2015). Nous commencerons par placer notre sujet dans son contexte en présentant son état de l'art du domaine. Nous présenterons ensuite en section 3 le corpus ANCOR qui nous a servi de support tout le long de notre travail. Le cadre expérimental de notre étude est détaillé en section 4. La section 5 énumère les traits linguistiques qui jouent un rôle prépondérant dans nos expérimentations, qui seront elles développées ensuite dans la section 6. Cette section 6 correspond à la part la plus original de notre travail : nous y étudions l'impact du degré d'interactivité des différents corpus sur le système ainsi qu'une stratégie de résolution à l'aide d'un multi-classifieur.

2. État de l'art

Les premières techniques de résolution de coréférence se sont tout d'abord basées sur des systèmes à base de règles (rule-based approach). Certains travaux comme ceux de (Hobbs, 1978) et (Lappin & Leass, 1994) se sont notamment concentrés sur le cas des coréférences pronominales. Ces approches se basaient sur une analyse syntaxique profonde des énoncés qui pose souvent des problèmes de robustesse, ce qui a conduit dans un premier temps à la proposition alternative d'approches purement heuristiques (Mitkov, 2002). L'analyse se limite dans ce cas à évaluer une fonction heuristique suivant la présence ou l'absence de traits le plus souvent locaux.

Ces traits se retrouvent, au tournant du millénaire, dans des travaux pionniers (Soon et al., 2001) et (Ng & Cardie, 2002) relevant de l'apprentissage supervisé sur corpus. Cette approche centrée sur les données, qui a été rapidement dominante, consiste à réaliser une classification binaire (coréférent / non coréférent) sur toutes les paires de mentions présentes dans le texte. Outre la mise à disposition

de corpus annotés de taille suffisante, elle nécessite toutefois un travail fin d'ingénierie sur les traits linguistiques d'apprentissage, travail dont permettent de s'affranchir les techniques neuronales d'apprentissage profond. À la suite de (Clark & Manning, 2016) et (Wiseman et al., 2016), puis (Lee et al., 2017) pour un traitement de bout en bout (end-to-end), les techniques connexionnistes connaissent ainsi un développement rapide sur cette problématique, où elles présentent de bonnes performances sur les données de campagnes d'évaluation telle que la *shared task* CoNNL'2012.

La communauté française a eu longtemps peu d'opportunité de conduire des travaux par apprentissage automatique du fait d'un manque crucial de ressources annotées en coréférence pour le français. C'est seulement en 2014, avec la création du corpus ANCOR – présenté ultérieurement – que les recherches sur le français ont pu prendre une nouvelle envergure. En dehors de deux systèmes à base de règles (Longo & Todirascu 2009 ; Godbert & Favre, 2017), nous pouvons aussi citer le projet européen SENSEI (Kabadjov & Stepano, 2016) où un système de résolution des coréférences a été développé par apprentissage sur le corpus ANCOR et intervient dans un processus d'analyse du discours. Notre travail se situe dans la continuité des travaux de (Désoyer et al., 2015) sur le système CROC : nous reprenons en effet l'idée d'un apprentissage supervisé à base de SVM, une approche de résolution *mention-pair* et enfin une reprise des traits descriptifs (*features*) retenus dans le système CROC. C'est à partir de cette base que nous avons proposé une nouvelle stratégie d'analyse reposant sur un multi-classifieur. Un de nos objectifs à terme est de poursuivre ce travail d'optimisation afin de définir une *baseline* robuste pour étudier l'apport réel des techniques d'apprentissage profond par rapport aux classifieurs de l'état de l'art. Il est à noter qu'à l'instar du travail de Désoyer, nous n'avons pas développé pour l'heure un système bout-en-bout. On suppose ici que les mentions sont déjà identifiées : ce sont celles qui sont présentes dans le corpus ANCOR.

3. Corpus ANCOR

Afin de palier au manque de ressource francophone annotée en coréférence, les laboratoires LIFAT et LLL ont réalisé le corpus ANCOR (Muzerelle et al., 2014 ou Desoyer et al., 2015), premier corpus français d'envergure répertoriant les relations de coréférence et relations anaphoriques. Il se compose de trois sous-corpus de parole transcrite qui présentent autant de situations de production orale, entre entretiens et dialogues interactifs (cf. tableau 1). Leur degré d'interactivité est par ailleurs variable et dépend de la situation discursive.

Corpus	Type de dialogue	Interactivité	Durée	Taille (mots)
ESLO_ANCOR	Interview	Faible	25 heures	417 000
OTG	Dialogue en face à face	Forte	2 heures	26 000
Accueil_UBS	Dialogue téléphonique	Forte	1 heure	10 000

TABLE 1 : Description des sous-corpus du corpus ANCOR

L'intérêt du corpus est de proposer une annotation riche qui peut servir à la fois à la conduite d'études linguistiques et à l'apprentissage automatique. Chaque mention, i.e. chaque entité linguistique référant à un objet du discours, et chaque relation sont décrites par un ensemble de propriétés. Une mention est ainsi décrite par l'ensemble de propriétés linguistiques suivant :

- Le Genre (masculin/féminin) et le Nombre (singulieur/pluriel) : GENRE et NB
- L'inclusion ou non dans un groupe prépositionnel : GP
- Type d'entité nommée (toponyme, anthroponyme…) : EN
- La définitude (Défini, indéfini, démonstratif ou explétif) : DEF
- Caractère générique ou spécifique de la référence de l'item considéré : GEN_REF
- Nouvelle entité du discours ou non : NEW

L'annotation a par ailleurs consisté à caractériser l'ensemble des relations de référence entre les mentions présentes dans le corpus. Une relation est quant à elle décrite par des indications sur l'accord en nombre et en genre des deux mentions reliées, mais également le type de la reprise. Cinq classes de relations sont ainsi définies, qui se partagent entre deux types principaux :

- Les **coréférences**, où les deux mentions réfèrent à la même entité du discours : on fait la distinction entre coréférence directe, indirecte, pronominale
- Les **anaphores associatives**, où les deux mentions ne sont pas coréférentes mais partagent un lien référentiel : nominale ou pronominale

Une relation directe correspond à une coréférence où la reprise et l'antécédent ont la même tête nominale ("*le petit garçon … ce gentil garçon*"). À l'inverse, une relation indirecte associera deux expressions aux têtes nominales disjointes, rapprochées généralement par un phénomène de synonymie, hyponymie ou bien d'hyperonymie ("*la maison … le bâtiment … la demeure*"). Enfin, la coréférence pronominale décrit une reprise par un pronom ("*le chien … il*").

Les relations associatives nominales relient deux mentions qui ne sont pas coréférentes. Néanmoins, ces mentions partagent une relation ontologique, c'est-à-dire que l'interprétation de l'une dépend de l'autre, par exemple suivant une relation méronymique ("*le gâteau … sa dernière part*"). La relation associative pronominale répond à cette définition mais met en jeu une reprise par un pronom ("*la France … ils ont gagné*"). Les travaux présentés ici ne concernent que la coréférence.

4. Cadre expérimental

Le travail qui est présenté dans cet article est une première tentative des laboratoires LIFAT et LIFO de construire un système de résolution des coréférences qui, à terme, sera utilisé entre autres pour la détection et le suivi de nominations[1], ceci dans le cadre du projet ANR TALAD. Comme base de réflexion, nous avons souhaité reproduire les expérimentations faites au laboratoire LATTICE

[1] Les nominations sont des formes de désignation émergentes, promues généralement par une communauté sociale donnée pour décrire un nouveau concept, et qui n'ont pas encore été figées dans la langue. Elles sont étudiées en particulier par la communauté d'analyse du discours.

(Desoyer et al. 2015) avec le système CROC. Dans un premier temps, nous avons donc reproduit une *baseline* assez proche de ces travaux. Nous nous poserons ensuite la question d'une amélioration des performances. Nous présenterons ici le cadre expérimental dans lequel a été conduite cette étude.

4.1. Techniques d'apprentissage

Les expérimentations que nous avons conduites ont été réalisées sur la plateforme Weka (Witten et al., 2011) qui fournit de nombreux algorithmes d'apprentissage automatique. Trois types de classifieurs ont été considérés : les arbres de décision (retenus pour le caractère explicatif du modèle de classification obtenu), les séparateurs à vaste marge (SVM, retenus pour le niveau reconnu de performance) et un classifieur bayésien naïf (Naive Bayes) comme *baseline*. Il est à noter que nous avons utilisé les paramètres par défaut de Weka pour chacun d'entre eux.

Le modèle d'arbre de décision j48 intégré dans Weka est une implémentation de l'algorithme C4.5 (Quinlan, 1993). Il construit par apprentissage supervisé un arbre où chaque noeud de décision (dit aussi nœud interne) représente un test sur un unique attribut. La sélection d'un test se fait par le choix de l'attribut qui discrimine au mieux les données et ainsi aura une meilleure qualité de séparation.

SVM est une méthode de classification binaire introduite par Vapnik et Chervonenkis et développée par (Boser et al., 1992). Elle consiste à déterminer une séparation par hyperplan de marge optimale dans l'espace multidimensionnel qui décrit les données d'apprentissage. Afin de traiter des problèmes non linéairement séparables, des fonctions noyaux permettent de transformer l'espace de représentation en un espace de plus grande dimension où l'on cherche une séparation linéaire. Nous avons étudié ici un classifieur linéaire, implémenté par la librairie LibSVM, et un classifieur polynomial avec la libraire SMO qui implémente l'algorithme d'optimisation de (Platt, 1998).

Enfin, Naive Bayes est un algorithme d'apprentissage statistique qui s'inspire du théorème de Bayes. Il repose sur l'hypothèse que chaque variable d'apprentissage est soumise à une indépendance conditionnelle. Cette indépendance supposée facilite l'estimation du modèle de classification, qui peut donner des performances correctes avec assez peu de données d'apprentissage.

4.2. Constitution de l'ensemble des corpus d'apprentissage et de test

Nous avons divisé ANCOR en différents sous-corpus équilibrés dans l'idée d'étudier l'influence de plusieurs paramètres sur la résolution de la coréférence. Nous avons évoqué précédemment le degré d'interactivité, variant d'un corpus à un autre. Nous avons considéré ce facteur de discrimination car cette interactivité semble avoir un fort impact sur la réalisation des chaînes de coréférences. Nous distinguerons donc des sous-corpus d'entraînement spécifiques aux sous-corpus ESLO (peu interactif) et OTG (interactif), en maintenant un équilibrage (50%/50%) entre les deux situations d'interaction. Le corpus UBS n'est pas inclus dans les ensembles d'entraînement du fait de sa taille réduite.

Corpus d'entraînement		ESLO	OTG	Total
SmallTrainingSet	COREF	1500	1500	3000
	NOT_COREF	1075	1075	2150
MediumTrainingSet	COREF	1500	1500	3000
	NOT_COREF	1917	1917	3834
BigTrainingSet	COREF	1500	1500	3000
	NOT_COREF	2617	2617	5234

TABLE 2 : Constitution des ensembles d'apprentissage à partir des sous-corpus ANCOR

L'autre facteur concerne l'équilibre des instances positives/négatives. On connaît l'influence de la prévalence d'une classe sur l'apprentissage des classifieurs. Suivant (Désoyer et al., 2015), nous avons défini trois corpus d'apprentissage (*Small*, *Medium* et *BigTrainingSet* dans le tableau 2) répondant à des ratios respectifs de une, deux et trois instances négatives pour une positive.

		ESLO	OTG	UBS	Total
TestSet_*i*	COREF	500	222	197	919
	NOT_COREF	700	311	276	1287

TABLE 3 : Constitution des ensembles de test à partir des sous-corpus ANCOR

Les corpus d'entraînement contiennent la même proportion d'instances provenant d'ESLO et OTG. Nous avons partagé de même notre corpus de test en plusieurs sous-corpus répondant à des degrés d'interactivité différents, afin d'étudier l'impact de cette dimension dialogique. Le corpus de test a été créé à partir des relations restantes pour chaque corpus, en prenant soin que tous les sous-corpus présentent le même ration d'instances positives (coréférence) et négatives. Compte tenu des relations à disposition, ce ratio a été fixé à 1 positive pour 1,4 négatives. Cette fois, c'est le corpus UBS qui est retenu comme corpus interactif. Ce corpus de test est par ailleurs partagé en 3 sous-corpus (TestSet_*i* dans le tableau 3 ci-dessous) pour permettre des études statistiques en significativité sur les résultats. Afin d'éviter tout sur-apprentissage, l'apprentissage a été réalisé par validation croisée avec 10 plis.

5. Traits linguistiques

Afin de s'assurer d'un bon niveau de performance, il est important de fournir au classifieur des traits linguistiques d'apprentissage pertinents. Pour nos travaux, nous avons choisi de reprendre les traits définis par (Désoyer et al., 2015), en excluant néanmoins ceux se rapportant à l'introduction d'un nouvel élément dans le discours (m1_NEW, m2_NEW et id_NEW). Ces attributs, présents dans le corpus annoté ANCOR, sont difficilement identifiables automatiquement sur du texte brut : leur estimation automatique reste un challenge aussi délicat que la résolution des coréférences elle-même. Nous allons décrire rapidement les attributs d'apprentissage subsistant.

5.1. Traits non-relationnels

Les traits non-relationnels servent à décrire chaque mention, indépendamment de celle à laquelle elle pourrait être liée. Chaque mention d'une paire se voit donc attribuer une étiquette, respectivement *m1* pour la première et *m2* pour la seconde, à qui on associe certains traits non relationnels. Grâce à l'annotation du corpus ANCOR, nous obtenons la catégorie syntaxique (`TYPE`), la détermination (`DEF`), le genre (`GENRE`), le nombre (`NOMBRE`) et le type d'entité nommée (`EN`) d'une mention.

5.2. Traits relationnels

Les traits relationnels ont pour but de comparer deux mentions d'une paire en observant leur forme, leurs attributs non-relationnels ou leur distance. Deux types de traits peuvent être distingués.

Les traits booléens vérifient généralement l'identité entre les valeurs de traits non relationnels des mentions en question. On peut par exemple citer le trait vérifiant l'accord en genre entre les deux mentions concernées, ou d'autres types d'information tels que l'identité de forme des chaînes de caractères (trait `ID_FORM`), l'inclusion strictement complète et contiguë de la plus petite chaîne de caractère dans la plus grande (`ID_SUBFORM`) ou bien l'inclusion au sens large des items d'une mention dans ceux de l'autre (`EMBEDDED`). Ainsi, si l'on considère l'exemple suivant *m1* = "*le tigre féroce et menaçant*" et *m2* = "*le tigre menaçant*", nous obtiendrons respectivement `ID_FORM = FALSE, ID_SUBFORM = FALSE et EMBEDDED = TRUE`.

Les autres traits sont décrits par un nombre entier ou réel. Il s'agit pour la plupart de distances spatiales ou lexicales telles que les distances dans le texte entre deux mentions en termes de nombres de mots (`DISTANCE_WORD`), de caractères (`DISTANCE_CHAR`), de mentions (`DISTANCE_MENTION`) ou bien même de tours de paroles (`DISTANCE_TURN`). Mais il peut également concerner des attributs qui apportent de nouvelles informations lexicales à savoir `INCL_RATE` et `COM_RATE` qui indiquent respectivement le taux d'inclusion de tokens et le taux de tokens communs. Appliqués à l'exemple précédent, nous obtenons `INCL_RATE = 1` et `COM_RATE = 2/5`.

6. Expérimentations

6.1. Recherche du meilleur classifieur

Les premières expériences que nous avons menées ont consisté à étudier la complexité du problème considéré en comparant le niveau de performances obtenues, après entraînement sur l'ensemble du corpus d'apprentissage, par un classifieur linéaire et un classifieur polynomial. Ces expérimentations ont été réalisées avec l'ensemble d'apprentissage *MediumTrainingSet* et les trois ensembles de test. Le tableau 4 contient la moyenne des F-mesures obtenues sur les trois sous-corpus. Nous évaluons ici la classification brute, c'est-à-dire la qualité de la classification `COREF/NOT_COREF` de chaque paire de mention, et non pas l'identification des chaînes (ou ensembles) de coréférence finales.

Techniques d'apprentissage	Moyenne de la f-mesure sur les trois TestSet_*i*
SVM polynomial (librairie SMO)	0,924
SVM linéaire (librairie LibSVM)	0,917

TABLE 4 : F-mesure en classification pure des SVM polynomial (SMO) et linéaire (LibSVM)

Le SVM polynomial dépasse légèrement SVM linéaire, avec un écart de 0,007 de f-mesure. Cette différence est toutefois significative d'un point de vue statistique (test de Wilcoxon-Mann-Withney : $Z_{inv} = 0{,}0633 < 0{,}1$). La faible amplitude de cette amélioration des performances pourrait laisser à penser que le problème de classification que nous considérons reste relativement simple. Il faut toutefois noter que ces résultats ont été obtenus sans recherche d'une optimisation du modèle polynomial construit. Par ailleurs, nous étudions ici la classification pure, qui donnera a priori des différences plus sensibles en termes d'identification finale des ensembles de coréférence complets. Nous avons donc considéré les performances du SVM polynomial suffisamment élevées pour le choisir comme représentant des séparateurs à vaste marge pour la suite.

	Small	Medium	Big
j48 (arbres de décision)	0,938	**0,946**	0,943
Naïve Bayes	**0,890**	0,887	0,873
SMO (SVM polynomial)	0,920	0,924	**0,929**
Moyenne	*0,916*	*0,919*	*0,915*

TABLE 5 : Influence du corpus d'apprentissage sur les performances en classification pure

Dans un second temps, nous avons étudié l'impact de l'équilibre entre instances positives et négatives lors de l'apprentissage, ceci avec les trois types classifieurs. La comparaison a donc porté sur les trois corpus d'apprentissage *Small-, Medium- et BigTrainingSet*). Le tableau 5 donne les performances, toujours en F-mesure de classification pure, de chaque classifieur. On constate que le ratio entre exemples positifs et négatifs a une influence modérée. Le corpus *MediumTrainingSet* est celui qui permet les meilleures performances en moyenne, mais ce résultat varie suivant le classifieur. C'est cet équilibrage moyen que nous conserverons dans le reste de notre étude, tout en relevant le peu d'impact de cette variable. Par ailleurs, nous constatons que j48 semble être le meilleur modèle avec une F-mesure de 0,946. À l'inverse, le faible résultat de Naive Bayes nous contraint à le délaisser, au profit du SVM qui reste relativement proche de j48 avec 0,929 de F-mesure maximale. Ces résultats confirment nos intuitions, fondées à la fois sur (Desoyer et al., 2015) et les limites connues des classifieurs bayésiens. Ils nous laissent penser que le corpus d'ANCOR présente une taille suffisante pour répondre aux besoins de l'apprentissage sur cette tâche, un classifieur bayésien étant connu pour être moins sensible au manque de données.

Nous avons enfin cherché à confirmer ces résultats sur les performances de résolution des chaînes (ou ensembles) complètes de coréférence. Nous avons pour cela utilisé les métriques MUC (Vilain

et al., 1995) et B^3 ((Bagga and Baldwin, 1998). Les ensembles de mentions coréférentes sont obtenus, dans toutes nos expériences, suivant une stratégie de type *best-first* à partir des résultats de la classification par paire : si une mention a plusieurs antécédents potentiels, on la place dans le premier ensemble de coréférence donné. La table 6 détaille les performances obtenues. On retrouve les résultats précédents sur la hiérarchie des classifieurs. De même, l'impact de l'équilibrage positif/négatif reste limité.

Métrique	Classifieur	Small	Medium	Big
MUC	j48	0,881	0,880	**0,891**
	NB	0,845	*0,854*	0,847
	SVM	0,854	0,859	*0,869*
	Moyenne	*0,860*	*0,864*	*0,869*
B^3	j48	**0,884**	0,874	0,876
	NB	0,855	0,847	*0,857*
	SVM	0,857	0,860	*0,864*
	Moyenne	*0,865*	*0,860*	*0,866*

TABLE 6 : Influence du corpus d'apprentissage sur les performances de j48 en résolution

6.2. Influence de l'interactivité des corpus

Le degré interactivité est très variable entre différentes situations de dialogue spontané, et peut conduire à des manifestations différentes de coréférence. Il nous est apparu important de vérifier dans quelle mesure les performances étaient influencées par ce degré d'interactivité. Pour cela, nous avons distingué dans les corpus d'apprentissage et de test des sous-corpus spécifiques à un degré d'interactivité donné (fort : OTG ou UBS ou faible : ESLO). Nous avons alors cherché à étudier si l'apprentissage de modèles spécifiques à chaque degré d'interactivité (suivant une approche par adaptation de modèles) ne pourrait pas conduire à de meilleurs performances qu'un modèle générique appris sur tout le corpus *MediumTrainingSet*,. Par exemple, pour un test en situation très interactive, nous distinguons :

- Modèle général - Apprentissage sur l'ensemble du corpus *MediumTrainingSet*,
- Adaptation de modèle - Apprentissage sur la sous-partie OTG de *MediumTrainingSet*

Nous avons étudié l'ensemble des différentes combinaisons *train*/*test* possibles. Les tables 7 et 8 décrivent les résultats obtenus respectivement par j48 et SMO dans quatre situations prototypiques. Nous avons rejoué 4 fois les expériences en changeant aléatoirement les instances négatives d'apprentissage, à fin d'étude en significativité statistique. L'évaluation porte ici sur la résolution complète (MUC et B^3), les résultats présentés étant cohérents avec ceux en classification pure.

Métrique de test	**MUC**		**B^3**	
Test (interactivité)	**UBS+OTG** (forte)	**ELSO** (faible)	**UBS+OTG** (forte)	**ELSO** (faible)
Modèle général	0,864 (σ = 0,007)	0,734 (σ = 0,010)	0,882 (σ = 0,006)	0,897 (σ = 0,004)
Modèle adapté	0,860 (σ = 0,003) (*train : OTG*)	0,742 (σ = 0,023) (*train : ESLO*)	0,884 (σ = 0,003) (*train : OTG*)	0,902 (σ = 0,009) (*train : ESLO*)

TABLE 7 : Influence du corpus d'apprentissage sur les performances en classification pure

Les observations montrent que l'impact du degré d'interactivité est réel. On observe ainsi une baisse de performances avec la mesure MUC entre les situations de faible interactivité et celles de forte interactivité. Avec j48, cette différence est statistiquement significative pour le modèle général ($|T| = 1,995 > T(0,1) = 1,983$) mais pas pour le modèle adapté ($|T| = 1,464 < T(0,1) = 1,983$). On remarque que la baisse observée avec MUC sur le corpus faiblement (ESLO) interactif est à l'opposé des résultats obtenus avec la mesure B^3 (où les différences ne sont pas significatives). On sait qu'une limitation de la mesure MUC est de ne pas prendre en considération les singletons (mentions ne faisant pas partie d'une chaîne de coréférence), contrairement à B^3. Une étude qualitative du comportement des modèles sur les singletons devra être menée pour expliquer ces résultats. Retenons pour l'heure que le degré d'interactivité peut influencer sur le comportement des systèmes.

On note ensuite que les modèles généraux entraînés sur l'ensemble du corpus *MediumTrainingSet* atteignent un niveau de performance équivalent à celui des modèles adaptés. Prenons l'exemple de j48. Dans le cas du différentiel de performances maximal observé avec ce classifieur (mesure MUC avec test sur ESLO : 0,734 contre 0,742), un test de Student donne une absence de toute différence statistiquement significative entre le modèle général et le modèle adapté : $|T| = 0,089 << T(0,1) = 1,983$. Les classifieurs généraux appris sur *MediumTrainingSet* semblent donc s'adapter par eux-mêmes à la diversité du degré d'interaction. Les gains de performance obtenus avec les modèles adaptés sont trop restreints pour justifier la construction de classifieurs spécifiques à chaque situation interactive. Pour cette raison, la suite de nos travaux concernera donc toujours des modèles génériques appris.

Métrique de test	**MUC**		**B^3**	
Test (interactivité)	**UBS+OTG** (forte)	**ELSO** (faible)	**UBS+OTG** (forte)	**ELSO** (faible)
Modèle général	0,847 (σ = 0,012)	0,729 (σ = 0,031)	0,868 (σ = 0,011)	0,992 (σ = 0,006)

Modèle adapté	0,845 (σ = 0,033)	0,717 (σ = 0,017)	0,893 (σ = 0,013)	0,897 (σ = 0,007)

TABLE 8 : Influence du corpus d'apprentissage sur les performances en résolution (j48)

6.3. Classifieurs spécifiques et multi-classifieur

Afin d'améliorer le niveau de performance de notre système, nous sommes partis de l'idée que les traits exploités n'étaient pas les mêmes en fonction du type de relation (e.g directe, indirecte et anaphore pronominale). Par exemple, dans un arbre de décision j48, le premier trait mis en valeur pour une relation directe sera `INCL_RATE` et `m2_TYPE` pour une relation indirecte. Nous avons alors cherché à savoir si la construction de classifieurs pour chaque type de coréférence ne pourrait pas conduire à une adaptation optimale des modèles sur chaque ensemble de traits d'apprentissage.

	Directe	**Indirecte**	**Anaphore**
F-mesure	0,973	0,969	0,965

TABLE 9 : F-mesure en classification pure (j48) des classifieurs spécifiques par relation

L'idée est donc de créer un classifieur spécifique pour chaque type de relation (i.e un classifieur de relation directe répondra uniquement `DIRECTE` ou `NOT_DIRECTE`). La table 9 semble indiquer que cette spécialisation est bénéfique, puisque les niveaux de performances en classification pure de chaque classifieur (expérience menée avec j48) sont très satisfaisants. Nous construisons ensuite un multi-classifieur, c'est-à-dire un système qui utilise les réponses de chaque classifieur comme vote pour la décision finale coréférente/non coréférente. Le système de vote n'est pas majoritaire : si l'un d'eux renvoie une réponse positive (i.e DIRECTE, INDIRECTE ou ANAPHORE) alors la relation est considérée coréférente. À l'inverse, elle sera jugée non-coréférente si tous renvoient une réponse négative (i.e NOT_DIRECTE, NOT_INDIRECTE, NOT_ANAPHORE). Les f-mesures obtenues par j48 de ce multi-classifieur sont comparées dans la table 10 avec celles du classifieur général original. Bien que parfois proches, les résultats de notre multi-classifieur demeurent inférieurs à ceux de celui de base. Il s'agit ici de travaux préliminaires qui demandent à être poursuivies avec le SVM.

	Classifieur de base	**Multi-classifieur**
Moyenne du testSet 1	0,9383	0,9333
Moyenne du testSet 2	0,9406	0,935
Moyenne du testSet 3	0,9353	0,933

TABLE 10 : F-mesure (classification pure) du classifieur originel et du multi-classifieur

7. Conclusion et perspectives

Dans cet article, nous avons présenté nos premières recherches visant à construire un système de résolution des coréférences basé sur des techniques de classification binaire (coréférent/non coréférent). Nous avons étudié une stratégie de résolution par multi-classifieur qui n'a pas été expérimentée à notre connaissance sur le français. Les résultats obtenus restent perfectibles, en particulier, nous poursuivons actuellement nos expérimentations sur l'ensemble des facteurs (choix des échantillons négatifs, optimisation des paramètres des classifications) pouvant influencer les paramètres. De même, il nous reste à élaborer un système de bout en bout travaillant sur des corpus bruts, en intégrant un détecteur de mentions développé au LATTICE (Grobol et al. 2017). Nous comptons par ailleurs poursuivre un travail d'ingénierie fine sur les traits d'apprentissage, tout en évaluant notre système sur les métriques CEAF (Luo 2005) et BLANC (Recassens & Hovy 2011).

Un de nos objectifs est de développer une *baseline* solide pour challenger les techniques d'apprentissage profond. Nous nous demandons en effet si, sur une tâche complexe comme la coréférence, l'intérêt des techniques neuronales réside sur leur niveau réel de performances ou sur le fait qu'elles dédouanent le chercheur d'un travail fastidieux d'ingénierie sur les traits linguistiques. Des études récentes montrent en effet que les techniques d'apprentissage présentent des performances très perfectibles (Durett & Klein, 2013) sur des tâches complexes telles que la résolution des coréférences pronominales ambigües ou les schémas de Winograd (Morgenstern et al., 2016), et que les approches neuronales n'ont pas surmonté cette difficulté.

Par ailleurs, l'objectif applicatif de nos travaux est de se focaliser sur la coréférence indirecte, dans une perspective de détection des variantes de nomination en analyse du discours. Il est à craindre qu'une approche neuronale puisse manquer de données d'apprentissage dans le cas des nominations, expressions en émergence et encore non figées dans la langue. Ceci reste à vérifier.

Remerciements

Ce travail s'inscrit dans le cadre de différents stages de fin de licence, encadrés par Jean-Yves Antoine, Anaïs Lefeuvre-Halftermeyer, Nicolas Labroche, Sylvie Billot et Marcilio de Souto que les auteurs tiennent à remercier. Cette recherche s'insère également dans le programme « Investissements d'Avenir » géré par l'Agence Nationale de la Recherche ANR-10-LABX-0083 (Labex EFL).

Références

BAGGA A., BALDWIN B. (1998). Algorithms for scoring coreference chains. *Proc. of the LREC Workshop on Linguistic Coreference*, pp. 563–566, Granada, Spain.

BOSER B. E., GUYON I., VAPNIK V. (1992) A training algorithm for optimal margin classifiers. *Proc. of the Fifth Annual Workshop on Computational Learning Theory*, ACM Press.

CLARK K., Manning C. D. (2016b). Improving coreference resolution by learning entity level distributed representations. In Association for Computational Linguistics (ACL).

DÉSOYER A., LANDRAGIN F., TELLIER I., LEFEUVRE A., ANTOINE J-Y. (2015). Les coréférences à l'oral : une expérience d'apprentissage automatique sur le corpus ANCOR. *Traitement Automatique des Langues*, *TAL*, vol. 55(2), pp.97-121.

DURRET G., KLEIN D. (2013) Easy victories and uphill battles in coreference resolution. *Proc. EMNLP'2013*.

GODBERT E., FAVRE B. (2017). Détection de coréférences de bout en bout en français. *Actes TALN'2017*, Orléans, Juin 2017.

GROBOL L., TELLIER I., DE LA CLERGERIE E., DINARELLI M., LANDRAGIN F. (2017) Apports des analyses syntaxiques pour la détection automatique de mentions dans un corpus de français oral. *Actes TALN 2017*, Orléans, France.

KABDJOV M., STEPANOV, J. (Eds.) (2016) The SENSEI Discourse Analysis Tools. *Rapport Technique SENSEI D4.2*.

LAPPIN S., LEASS H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20, pp. 535–561.

LEE K., HE L., LEWIS M., ZETTLEMOYER L. (2017) End-to-end neural coreference resolution. *Proc. EMNLP'2017*.

LONGO L., TODIRASCU A. (2009). Une étude de corpus pour la détection automatique des thèmes, *Actes des 6èmes journées de linguistique de corpus*, Lorient. 10-12 septembre 2009.

LUO X. (2005). On coreference resolution performance metrics. *Proc. HLT-EMNLP 2005*, pp. 25–32, Vancouver, Canada.

MITKOV R. (2002). *Anaphora resolution*. Longman.

MORGENSTERN L., DAVIS E., ORTIZ C.L. (2016) Planning, executing and evaluating the Winograd Schema Challenge. *AI Magazine*, 37(1). Pp. 50-54.

MUZERELLE J., LEFEUVRE A., SCHANG E., ANTOINE J-Y., PELLETIER A., MAUREL D., IESHKOL I., VILLANEAU J. (2014). ANCOR_CENTRE, a large free spoken French coreference corpus : description of the resource and reliability measures. *Proc. LREC'2014*, Reykjavik, Islande.

NG V., CARDIE C. (2002). Improving machine learning approaches to coreference resolution. In Proceedings of the ACL. *Proc.* of the *ACL'02*. pp. 104-111.

PLATT J. (1998) Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*.

QUINLAN J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

RECASENS, M. HOVY, E. (2011). BLANC: Implementing the Rand index for coreference evaluation. *Natural Language Engineering*, 17(04), pp. 485 - 510

SOON W., NG H., LIM D. (2001). A Machine Learning Approach to Coreference of Noun Phrases. *Computational Linguistics*, 27(4), 521-554.

VILAIN M., BURGER J., ABERDEEN J., CONNOLLY D., HIRSCHMAN L. (1995). A model-theoretic coreference scoring scheme. *Proc. MUC-6 Conference*, pp. 45–52

WIDLÖCHER A., MATHET Y. (2009). La plate-forme Glozz : environnement d'annotation et d'exploration de corpus. *Actes TALN'2009*. Senlis, France.

WISEMAN S., RUSH A.M., SHIEBER S.M. (2016) Learning global features for coreference resolution. *Proc. Human Language Technology and North American Association for Computational Linguistics, HLT-NAACL'2016*.

WITTEN I.H., EIBE F., HALL M.A. (2011). *Data Mining: Practical machine learning tools and techniques*,3e édition, Morgan Kaufmann.

Approche lexicale de la simplification automatique de textes médicaux

Rémi Cardon
UMR 8163 – STL – Savoirs Textes Langage, F-59000 Lille, France
remi.cardon@etu.univ-lille3.fr

RÉSUMÉ
Notre travail traite de la simplification automatique de textes. Ce type d'application vise à rendre des contenus difficiles à comprendre plus lisibles. À partir de trois corpus comparables du domaine médical, d'un lexique existant et d'une terminologie du domaine, nous procédons à des analyses et à des modifications en vue de la simplification lexicale de textes médicaux. L'alignement manuel des phrases provenant de ces corpus comparables fournit des données de référence et permet d'analyser les procédés de simplification mis en place. La substitution lexicale avec la ressource existante permet d'effectuer de premiers tests de simplification lexicale et indique que des ressources plus spécifiques sont nécessaires pour traiter les textes médicaux. L'évaluation des substitutions est effectuée avec trois critères : grammaticalité, simplification et sémantique. Elle indique que la grammaticalité est plutôt bien sauvegardée, alors que la sémantique et la simplicité sont plus difficiles à gérer lors des substitutions avec ce type de méthodes.

ABSTRACT
Lexical approach for the automatic simplification of medical texts

Our work addresses the automatic text simplification. This kind of application aims at improving the readability of texts that are difficult to read. Using three different corpora – which contain biomedical texts – an existing lexicon and a domain terminology, we perform analysis and modification of texts in order to achieve their lexical simplification. Manual alignment of sentences from comparable corpora provides reference data and permits to analyze the simplification procedures involved. Lexical substitution using existing resources permits to perform first tests of lexical simplification and indicates that specific resources are necessary when working with medical contents. The evaluation of substitutions is performed through three criteria : grammaticality, simplicity and semantics. It indicates that grammaticality is rather well preserved, while semantics and simplicity are more difficult to handle during the substitutions with this kind of methods.

MOTS-CLÉS : Simplification automatique de textes, analyse lexicale, domaine médical, simplification lexicale, substitution lexicale.

KEYWORDS: Automatic text simplification, lexical analysis, medical area, lexical simplification, lexical substitution.

1 Introduction

La simplification automatique de textes est un domaine du TAL, dans lequel il s'agit d'appliquer des transformations sur les phrases d'un texte pour les rendre plus lisibles, tout en conservant leur sens

intact. Cela est pratiqué aussi bien à destination des humains que pour faciliter les tâches nécessitant l'analyse automatique de textes (Chandrasekar *et al.*, 1996). La simplification, dont l'objectif est de faciliter des traitements d'analyse automatique, peut faire partie de différentes applications. Ainsi, la première application à l'avoir exploitée cherchait à simplifier les structures de phrases avant de procéder à leur analyse syntaxique automatique (Chandrasekar *et al.*, 1996). Dans d'autres contextes, la simplification peut être utilisée pour adapter certains genres de textes à des outils, qui n'ont pas été entraînés pour les traiter spécifiquement, comme par exemple l'analyse d'un texte biomédical effectuée avec des outils entraînés sur des textes journalistiques (Jonnalagadda *et al.*, 2009). Pour la simplification à destination des humains, ces méthodes sont explorées pour différents objectifs et différents publics. Notons par exemple que la simplification est effectuée pour des personnes avec une faible compétence de lecture (Williams & Reiter, 2005), pour des personnes sourdes qui montrent des difficultés de lecture et d'écriture (Inui *et al.*, 2003), pour des lecteurs dyslexiques (Rello *et al.*, 2013) ou encore pour des personnes autistes (Barbu *et al.*, 2013). Dans le domaine médical – dans lequel nous nous plaçons ici – la simplification peut également servir à faciliter l'éducation thérapeutique des patients (Brin-Henry, 2014) ou l'accès à l'information par les enfants (De Belder & Moens, 2010). En effet, des études ont montré qu'une meilleure compréhension des informations de santé par les patients et leurs familles mène à une meilleure adhésion au traitement et à un processus de soins plus réussi (Davis & Wolf, 2004; Berkman *et al.*, 2011).

L'objectif de notre travail consiste à contribuer au domaine de la simplification de textes de spécialité, sur l'exemple de textes médicaux. D'une part, nous proposons de constituer des corpus comparables différenciés par leur spécialisation, d'effectuer un alignement de phrases à partir de ces corpus afin de faire une analyse des procédés de simplification mis en place. D'autre part, nous proposons d'effectuer de premiers tests de simplification lexicale en utilisant la substitution. Si la majorité de travaux de simplification traitent les données en langue anglaise, nous travaillons avec les données en français.

Dans la suite de ce travail, nous présentons d'abord l'état de l'art (section 2), ensuite les données sur lesquelles nous travaillons (section 3). La méthode est présentée dans la section 4 et les résultats dans la section 5. Nous proposons une conclusion et les perspectives de ce travail dans la section 6.

2 État de l'art

Les travaux en simplification automatique se positionnent essentiellement à deux niveaux : lexical et syntaxique. La *simplification lexicale* opère au niveau des unités lexicales. Nous allons illustrer la simplification lexicale avec la tâche de simplification proposée lors de la compétition *SemEval 2012* [1] pour la langue anglaise. Pour un texte court et un mot cible, plusieurs substitutions possibles et satisfaisant le contexte ont été proposées par les organisateurs. L'objectif consistait à trier ces substitutions selon leur degré de simplicité (Specia *et al.*, 2012) et donc de les positionner les unes par rapport aux autres, en fonction de leur difficulté. Par exemple, pour la phrase *Hitler committed terrible atrocities during the second World War*, le mot à substituer est *atrocities*. Les candidats synonymes proposés par les organisateurs sont *abomination, cruelty, enormity, violation*. Le choix de référence est *cruelty*, ce qui doit produire en sortie : *Hitler committed terrible cruelties during the second World War.* De manière générale, lors de la simplification lexicale, plusieurs étapes peuvent être distinguées :

1. L'identification de mots ou termes qui peuvent poser des difficultés de compréhension. Cette étape est le plus souvent accomplie à l'aide de ressources lexicales auxquelles sont associées

1. http://www.cs.york.ac.uk/semeval-2012/

des mesures de complexité des mots, même si ces mesures n'ont pas reçu le consensus de la communauté de recherche (Saggion, 2017; Shardlow, 2014). Comme indiqué plus bas, des mesures classiques et computationnelles, et donc plus récentes, sont distinguées ;

2. Le remplacement de ces unités par un équivalent jugé plus facile de compréhension. Cette étape repose sur la disponibilité d'un dictionnaire d'expressions synonymiques (Shardlow, 2014) ou même hyperonymiques ;
3. Lorsque plusieurs équivalents sont disponibles, il est nécessaire de les ordonner par rapport à leur niveau de difficulté pour être en mesure de sélectionner les candidats les plus faciles à comprendre (François *et al.*, 2016). C'était typiquement la tâche proposée lors de la compétition *SemEval 2012*. Les participants ont exploité plusieurs critères pour effectuer cette tâche : lexique d'un corpus oral et de Wikipedia, n-grammes de Google, WordNet (Sinha, 2012) ; longueur de mots, nombre des syllabes, information mutuelle, fréquences (Jauhar & Specia, 2012) ; fréquences dans Wikipedia, longueur de mots, n-grammes, complexité syntaxique des documents (Johannsen *et al.*, 2012) ; n-grammes, fréquences dans Wikipedia, n-grammes de Google (Ligozat *et al.*, 2012) ; WordNet, fréquences (Amoia & Romanelli, 2012) ;
4. Il faut également s'assurer que les candidats à substitution sont acceptables dans le contexte de chaque phrase traitée. Dans *SemEval 2012*, cette condition était assurée par les organisateurs.

La *simplification syntaxique* opère au niveau de la syntaxe. Son objectif est de réorganiser la structure syntaxique des phrases. Quelques exemples d'opérations de réorganisation de cette structure sont : le découpage de phrases complexes en plusieurs phrases plus simples, l'ajout ou la suppression de propositions, la modification de temps verbaux (Brouwers *et al.*, 2014; Gasperin *et al.*, 2009; Seretan, 2012). Pour un exemple, la phrase *Elle compte, selon les autorités du pays, en 2009, 5 878 609 habitants pour l'agglomération, et 3 796 677 habitants pour la ville, soit 20 % de la population totale du pays* devient, après l'application d'une règle de suppression de propositions subordonnées et d'incises : *Elle compte, selon les autorités du pays, en 2009, 5 878 609 habitants pour l'agglomération, et 3 796 677 habitants pour la ville* (Brouwers *et al.*, 2014).

Pour ces deux types de simplification, des approches à base de règles et de probabilités ont été développées. Les approches à base de règles reposent sur l'expertise des concepteurs et leur connaissance des procédés de simplification. Notons que cette expertise peut également profiter d'un corpus de simplification déjà disponible. Les approches statistiques nécessitent de disposer de corpus de textes comparables, ou même parallèles et alignés, qui contiennent typiquement des textes complexes et leurs versions simplifiées (Zhu *et al.*, 2010; Specia, 2010; Woodsend & Lapata, 2011). Un des exemples typiques de corpus comparables, qui sont largement utilisés dans ce type de travaux, sont Wikipedia [2] et Simple Wikipedia [3] en langue anglaise. Si le premier propose des articles à destination de la population en général, Simple Wikipedia vise des populations spécifiques (comme par exemple les enfants, les apprenants d'anglais, les adultes en difficulté de lecture, etc.).

Une des tâches importantes de la simplification automatique consiste à pouvoir mesurer la lisibilité des unités linguistiques. Différents niveaux de mesure sont utilisés, que ce soit au niveau des termes ou des textes. Deux grands types de mesures de lisibilité peuvent être distingués : classiques et computationnelles (François, 2011). Les mesures classiques reposent sur le calcul de la complexité de surface des mots (nombre de syllabes) et de phrases pour évaluer leur lisibilité. Par exemple, si les mots et les phrases d'un texte sont longs, il est considéré être difficile à lire (Flesch, 1948; Gunning, 1973; Dubay, 2004). Les mesures computationnelles fournissent la possibilité d'associer les

2. https://en.wikipedia.org/wiki/Main_Page
3. https://simple.wikipedia.org/wiki/Simple_English_Wikipedia

unités, dont on souhaite mesurer la lisibilité (mots, phrases, textes...), à une variété de descripteurs : combinaisons de mesures classiques avec des informations terminologiques (Kokkinakis & Gronostaj, 2006), utilisation de *n-grammes* de caractères (Poprat *et al.*, 2006), descripteurs discursifs (Goeuriot *et al.*, 2007), descripteurs morphologiques (Chmielik & Grabar, 2011), etc. En général, ces mesures sont calculées de manière supervisée par rapport à une référence et fournissent des résultats fiables.

Par rapport aux travaux existants, nous nous positionnons au niveau de la simplification lexicale. Nous proposons d'effectuer la simplification en effectuant des substitutions lexicales grâce à l'exploitation d'un lexique existant. Notre tâche est assez proche de celle proposée lors de *SemEval 2012*.

3 Présentation des données

Les données exploitées sont de deux types : les corpus comparables provenant du domaine médical, et les ressources utilisées pour l'analyse et la substitution lexicale.

3.1 Corpus

3.1.1 Cochrane

Cochrane [4] est un organisme qui a pour objectif la diffusion de l'information médicale (Sackett *et al.*, 1996). Les textes publiés par Cochrane sont des synthèses de la littérature médicale sur une question spécifique (diagnostic, traitement). Ces synthèses sont créées à destination des professionnels de santé. Plus récemment, elles sont simplifiées par les collaborateurs de Cochrane pour les rendre accessibles au grand public également. De même, les synthèses écrites en anglais sont traduites en d'autres langues, y compris le français. Le corpus que nous utilisons est composé de 3 815 synthèses en français à destination des médecins et, pour chacune d'elle, sa version simplifiée (voir la figure 1).

Version technique	*Version simplifiée*
L'hématome aigu de l'oreille est une affection qui se caractérise par la formation d'une collection sanguine sous le périchondre du pavillon. Il est souvent provoqué par un traumatisme contondant. En l'absence de traitement, il finit par entraîner une difformité couramment appelée oreille en chou-fleur ou oreille du boxeur.	L'hématome aigu de l'oreille est une affection qui se caractérise par la formation d'une collection sanguine dans le pavillon (oreille externe), souvent à la suite d'un traumatisme contondant. S'il n'est pas traité, il entraîne une difformité appelée oreille en chou-fleur ou oreille du boxeur.

FIGURE 1 – Exemple de textes comparables du corpus Cochrane.

3.1.2 Encyclopédie

Wikipedia [5] est une encyclopédie collaborative en ligne. Elle propose les informations à destination d'un large public et aborde un grand nombre de sujets. Vikidia [6] est également une encyclopédie

4. https://www.cochranelibrary.com
5. https://fr.wikipedia.org
6. https://fr.vikidia.org/

collaborative en ligne, sa spécificité est qu'elle est destinée aux enfants de 8 à 13 ans. Nous exploitons ces deux sources pour constituer un deuxième corpus de travail. Il est composé de 575 articles relatifs au portail de la Médecine de Wikipedia et des articles équivalents de Vikidia (voir la figure 2).

Version technique	*Version simplifiée*
La luette ou uvule est un appendice conique situé au fond de la cavité buccale et proche des tonsilles palatines. Le mot uvule vient du latin uva, qui signifie "grain de raisin". La luette est un organe de 10 à 15 millimètres de long, de forme tubulaire quand il est détendu, qui pend à la partie moyenne du bord inférieur du voile du palais. Elle est constituée d'un tissu membraneux et musculaire.	La luette ou uvule est un appendice conique située au fond de la bouche. C'est un organe fait de tissus membraneux et musculaires, d'environ 10 à 15mm de long, qui pend à la partie moyenne du voile du palais.

FIGURE 2 – Exemple de textes comparables du corpus Wikipedia/Vikidia.

3.1.3 Médicaments

Le troisième corpus contient les informations sur les médicaments issues de la base de données [7] du ministère de la santé. On peut accéder, pour un médicament donné, au résumé des caractéristiques du produit (RCP), créé à destination des professionnels, et à la notice, créée à destination du grand public. Ces dernières peuvent aussi être trouvées dans les boîtes de médicaments. Ce corpus contient 11 800 RCP techniques et leurs notices grand public (voir la figure 3).

Version technique	*Version simplifiée*
- hypersensibilité à l'huile de paraffine	- si vous avez une allergie à l'huile de paraffine
- colopathie obstructive, compte tenu de l'effet laxatif du médicament	- si vous êtes atteint de colopathie obstructive, compte tenu de l'effet laxatif du médicament
- syndrome douloureux abdominal de cause indéterminée et inflammatoire (rectocolite ulcéreuse, maladie de Crohn)	- si vous êtes atteint de syndrome douloureux abdominal de cause indéterminée et inflammatoire (rectocolite ulcéreuse, maladie de Crohn, ...)
- ne pas utiliser chez les personnes présentant des difficultés de déglutition en raison du risque d'inhalation bronchique et de pneumopathie lipoïde	- ne pas utiliser chez les personnes présentant des difficultés pour avaler en raison du risque d'inhalation de la paraffine liquide qui entraîne une pneumopathie lipoïde

FIGURE 3 – Exemple de textes comparables du corpus Médicament.

3.1.4 Bilan des corpus

Le tableau 1 fait le bilan des trois corpus et indique, pour chaque corpus : sa taille en nombre de documents, d'occurrences de mots et de lemmes uniques. Il s'agit de corpus comparables : les articles

7. https://base-donnees-publique.medicaments.gouv.fr/

concernent les mêmes sujets mais relèvent de différents discours. Comme le montrent les extraits, ils proposent rarement une réécriture de la version technique (originale) en langage simplifié. En effet, le plus souvent, la création de la version simplifiée semble être indépendante de la version technique. Nous pouvons voir que le corpus *Médicaments* est le plus gros des trois corpus étudiés, tandis que le corpus *Encyclopédie* est le plus petit. Par ailleurs, les versions techniques sont toujours plus volumineuses que les versions simplifiées.

Corpus	*nb doc.*	*nb occ.*	*nb lemmes uniques*
Cochrane technique	3 815	2 804 336	11 558
Cochrane simplifié	3 815	1 491 243	7 567
Médicaments technique	11 800	51 705 111	43 515
Médicaments simplifié	11 800	33 116 119	25 725
Wikipedia	575	2 186 891	19 287
Vikidia	575	183 051	3 117

TABLE 1 – Taille des corpus comparables.

3.2 Ressources

Nous utilisons deux ressources : une terminologie spécialisée, Snomed International, et un lexique généraliste issu du Wiktionary.

3.2.1 Terminologie médicale Snomed

Nous exploitons la terminologie médicale Snomed International (Côté, 1996) telle que diffusée par ASIP santé. [8] La vocation de cette terminologie est de décrire le domaine médical. La terminologie contient 151 104 termes médicaux structurés en onze axes sémantiques (maladies et anomalies, actes médicaux, produits chimiques, organismes vivants, anatomie). Selon notre hypothèse, le contenu de cette terminologie permet d'estimer la couverture en termes et mots médicaux d'un texte donné.

3.2.2 Lexique issu du Wiktionary

Nous utilisons un lexique obtenu à partir des articles du Wiktionary [9], dans sa version GLAWI (Sajous & Hathout, 2015). Nous retenons les entrées dont au moins une définition est associée à l'une des catégories suivantes : *anat*, *chirurgie*, *génétique*, *maladie(s)*, *médecine*, *médicaments*, *microbiologie*, *neurologie*, *pathologie*, *pédologie*, *pharmacologie*, *physiologie*, *squelette*, *virologie*. Cela fournit un lexique avec 8 012 entrées uniques qui peuvent être liées au domaine médical. Les catégories les plus importantes sont *médecine* avec 4 967 entrées et *anat* (pour *anatomie*) avec 1 925 entrées. Les autres catégories contiennent entre quelques dizaines et quelques centaines d'entrées.

Cette ressource fournit des séries de synonymes et des hyperonymes pour certains termes. Ainsi, 25 % des entrées de GLAWI ont au moins un synonyme, avec une moyenne de 2,42 synonymes par entrée.

8. http://esante.gouv.fr/services/referentiels/referentiels-d-interoperabilite/snomed-35vf

9. https://fr.wiktionary.org/

Seules 452 entrées ont des hyperonymes : 318 avec un hyperonyme et 134 avec 2 à 6 hyperonymes. Les séries de synonymes et d'hyperonymes comportent le plus souvent des mots de la même partie du discours. Cette ressource contient essentiellement les entrées simples, mais nous avons également 1 157 entrées polylexicales. Enfin, cette ressource offre le paradigme flexionnel (le lemme et ses formes fléchies) des entrées.

4 Méthode

4.1 Pré-traitements

Les corpus sont pré-traités : l'étiquetage morpho-syntaxique et la lemmatisation sont effectués avec le TreeTagger (Schmid, 1994). Cela permet de normaliser le contenu des corpus grâce à la lemmatisation. Les termes de la terminologie Snomed International sont projetés sur les textes, ce qui permet d'effectuer une analyse lexicale de ces corpus.

4.2 Alignement de phrases

Nos corpus sont des corpus comparables. Notre objectif est de constituer une base de phrases alignées, qu'elles soient parallèles ou comparables, à partir desquelles nous pourrions faire des observations sur les manières dont la simplification peut être effectuée.

Une sélection aléatoire d'articles a fourni : 2*13 documents du corpus *Cochrane*, 2*12 documents du corpus *Médicaments* et 2*14 documents du corpus *Encyclopédie*. Ces articles sont utilisés pour effectuer l'alignement manuel de phrases provenant des corpus techniques et simplifiés. Deux annotateurs ont effectué l'alignement de manière indépendante. Des séances de consensus ont permis ensuite de résoudre les désaccords. Les critères qui guidaient l'alignement sont les suivants :

1. Les deux phrases doivent avoir le même sens ou sinon des sens proches, comme dans ces phrases du corpus *Cochrane* :
   — *les sondes gastriques sont couramment utilisées pour administrer des médicaments ou une alimentation entérale aux personnes ne pouvant plus avaler*
   — *les sondes gastrique sont couramment utilisées pour administrer des médicaments et de la nourriture directement dans le tractus gastro-intestinal (un tube permettant de digérer les aliments) pour les personnes ne pouvant pas avaler*
2. Le sens d'une phrase peut se retrouver intégralement dans le sens de l'autre phrase. Dans l'idéal, il devrait s'agir de l'inclusion sémantique, comme dans cet exemple, où la phrase technique indique le nombre de participants et la mesure d'évaluation en plus :
   — *peu de données (43 participants) étaient disponibles concernant la détection d'un mauvais placement (la spécificité) en raison de la faible incidence des mauvais placements*
   — *cependant, peu de données étaient disponibles concernant les sondes placées incorrectement et les complications possibles d'une sonde mal placée*
3. Les cas d'intersection sémantique, où chaque phrase apporte des informations spécifiques, seraient à proscrire. L'exemple du corpus *Cochrane* qui suit illustre ce cas :
   — *des études à plus grande échelle sont nécessaires pour déterminer la possibilité d'événements indésirables lorsque les ultrasons sont utilisés pour confirmer le positionnement des sondes*

— *des études à plus grande échelle sont nécessaires pour déterminer si les ultrasons pourraient remplacer les rayons x pour confirmer la mise en place d'une sonde gastrique, et pour évaluer si les ultrasons pourraient permettre de réduire les complications graves, telles que la pneumonie résultant d'un tube mal placé*

Les cas d'intersection sémantique sont plus difficiles à généraliser et à reproduire.

Dans ces exemples, nous voyons que le passage d'une phrase technique vers sa version simplifiée requiert des modifications syntaxiques et lexicales. Dans notre travail, nous nous concentrons sur la simplification lexicale effectuée au moyen de substitutions lexicales.

4.3 Substitution lexicale

Nous proposons d'aborder la simplification au niveau lexical grâce aux substitutions de mots par leurs équivalents supposés être plus simples et faciles à comprendre. Notre approche est fondée sur des règles. Elle suit les étapes suivantes :

— Les phrases des corpus techniques sont exploitées pour effectuer les tests de substitution ;
— L'ensemble médical du lexique Wiktionary est exploité pour fournir les candidats à substitution. Ce lexique propose en effet des ensembles de synonymes et d'hyperonymes ;
— Les entrées de ce lexique sont filtrées. Comme nous l'avons vu, le corpus *Vikidia* propose le contenu le moins spécialisé par rapport au reste des corpus. Nous projetons donc les entrées du lexique et leurs synonymes et hyperonymes sur le corpus *Vikidia*. Si une entrée, ses synonymes ou hyperonymes, sont reconnues dans ce corpus, nous supposons qu'il s'agit des entrées plus faciles à comprendre, et les retenons pour effectuer les substitutions lexicales ;
— Si une phrase contient une entrée du lexique qui possède des synonymes ou hyperonymes, si cette entrée ne se trouve pas dans le corpus *Vikidia*, et si un de ses synonymes ou hyperonymes se trouve dans le corpus *Vikidia*, ce synonyme ou hyperonyme est utilisé pour la substitution.

Il s'agit d'une méthode souvent exploitée dans les travaux de l'état de l'art, qui visent à effectuer la simplification lexicale des textes (Biran *et al.*, 2011; Wubben *et al.*, 2012; Horn *et al.*, 2014; Glavas & Stajner, 2015; Abualhaija *et al.*, 2017).

4.4 Évaluation

Les résultats de la substitution lexicale sont évalués avec plusieurs critères exploités dans les travaux existants en simplification lexicale (Biran *et al.*, 2011; Wubben *et al.*, 2012) :

— *Grammaticalité*. Le jugement sur la grammaticalité doit répondre à la question de savoir si la phrase reste grammaticale après les modifications effectuées. Pour assurer le respect de ce critère, lors de la simplification lexicale par exemple, la plupart des travaux effectuent la substitution avec des mots de la même catégorie syntaxique que le mot substitué (Biran *et al.*, 2011; Horn *et al.*, 2014; Glavas & Stajner, 2015; Abualhaija *et al.*, 2017) ;
— *Sémantique*. Le jugement sur la sémantique doit répondre à la question de savoir si la transformation effectuée préserve la sémantique originale de la phrase. En effet, quelle que soit la simplification effectuée, la sémantique des textes doit rester préservée ;
— *Simplicité*. Le jugement sur la simplicité doit répondre à la question de savoir si la simplification effectuée sur une phrase la rend plus simple à comprendre.

Ces critères sont évalués manuellement par l'auteur.

5 Résultats et leur discussion

Nous présentons les résultats relatifs à l'analyse lexicale des corpus (section 5.1), à l'alignement de phrases (5.2) et à la substitution lexicale en vue de simplification (section 5.3).

5.1 Analyse lexicale

L'analyse lexicale, basée sur la terminologie Snomed International, permet de calculer : (1) le nombre de ses mots et termes qui apparaissent dans chacun des corpus, et (2) le ratio d'occurrences des termes de la Snomed entre les textes en version technique et simplifiée. Notons que la lemmatisation avec TreeTagger peut empêcher la reconnaissance d'expressions polylexicales présentes dans la terminologie, comme {*bilirubine totale* ; *bilirubine total*} ou {*amnésie passagère* ; *amnésie passager*}. L'objectif de la projection ici est d'obtenir une estimation du degré de spécialisation de chaque sous-corpus, et plus précisément de pouvoir comparer ces estimations entre elles. Nous n'appliquons pas de traitement spécifique pour le repérage des expressions polylexicales.

Corpus	*Termes*
Cochrane simplifié	2 316
Cochrane technique	2 505
Médicaments simplifié	2 700
Médicaments technique	3 332
Encyclopédie simplifié	1 635
Encyclopédie technique	3 999

TABLE 2 – Nombre de termes uniques de la Snomed International dans les corpus.

Ratio	*Valeur*
Cochrane : simplifié / technique	0.60
Cochrane : technique / simplifié	1.66
Médicaments : simplifié / technique	0.62
Médicaments : technique / simplifié	1.62
Encyclopédie : simplifié / technique	0.10
Encyclopédie : technique / simplifié	9.67

TABLE 3 – Ratio des occurrences de termes de la Snomed International dans les corpus.

Les tableaux 2 et 3 présentent les résultats. Selon le tableau 2, nous trouvons plus de termes de la Snomed dans les versions techniques. Par ailleurs, le corpus Vikidia est le plus pauvre en termes spécialisés. Il s'agit certainement du corpus dont le niveau de lisibilité est le plus élevé. Ces observations sont corroborées par les indications du tableau 3 qui indique les ratios de termes entre les corpus techniques et spécialisés. Ces ratios sont comparables pour les corpus *Cochrane* et *Médicaments*. En revanche, nous observons une différence beaucoup plus importante entre les versions technique et simplifiée du corpus encyclopédique, où les versions simplifiées d'articles contiennent beaucoup moins de termes spécialisés.

5.2 Alignement de phrases

Corpus	*nb doc.*	*nb phrases total*	*nb phrases alignées*	*ratio*
Cochrane	26	653	240	36,75%
Médicaments	24	7101	258	3,63%
Encyclopédie	28	2651	163	6,15%

TABLE 4 – Nombre de phrases alignées pour chaque corpus.

Le tableau 4 indique les résultats consensuels de l'alignement manuel des phrases. Nous pouvons observer que les phrases alignées, qui font la correspondance entre le contenu technique et simplifié, sont relativement plus rares pour les corpus *Médicaments* et *Encylopédie*, alors que le corpus *Cochrane* en offre plus par rapport à sa taille. Les raisons de cet état de chose peuvent être les suivantes :

— La ligne directrice de rédaction des versions simplifiées des résumés de la fondation Cochrane affiche explicitement une volonté de simplifier le contenu de ses résumés d'origine pour le grand public. Les rédacteurs et traducteurs prennent donc comme point de départ les résumés originaux et techniques et les simplifient au fur et à mesure de l'avancement ;

— Pour les deux autres corpus, les principes ne sont pas aussi stricts. Ainsi, l'objectif de Vikidia est de traiter des sujets présents dans Wikipedia mais pour un public d'enfants. La création d'articles de Vikidia est rarement basée sur les articles de Wikipedia : le plus souvent, il s'agit d'une écriture indépendante. Quant au corpus *Médicaments*, les mêmes médicaments sont décrits et spécifiés dans les versions technique et simplifiée. Cependant, certaines informations sur les médicaments sont propres aux RCP (composition plus détaillée, action sur l'organisme, molécules, détail sur les effets indésirables...), alors que d'autres informations sont propres aux notices destinées au grand public (précautions d'emploi, mises en garde...).

Une analyse de ces phrases alignées nous indique aussi que les procédés de simplification (lexicale, syntaxique et stylistique) ne sont pas les mêmes selon les corpus :

— *Simplification lexicale.* Dans Vikidia, les notions complexes sont plutôt explicitées, alors que dans les corpus *Médicaments* et *Cochrane*, les notions complexes sont souvent suivies par leurs équivalents entre parenthèses :

  — *en revanche, les ultrasons associés à d'autres tests (par exemple, la visualisation de l'irrigation saline (injecter une solution saline à travers la sonde et l'observer à l'intérieur de l'estomac par ultrasons)) pourraient être utiles pour confirmer le placement des tubes utilisés pour le drainage gastrique*

  — *l'alimentation offerte au travers d'un tube placé par erreur dans la trachée (un conduit où passe l'air respiré) peut entraîner une pneumonie grave (une infection des poumons)*

  Quel que soit le corpus, les notions complexes peuvent aussi être remplacées par leurs équivalents plus simples. En voilà quelques exemples au format {*technique* ; *simplifié*} :

  {*alimentation* ; *nourriture*}, {*entérale* ; *directement dans le tractus gastro-intestinal*}, {*fournir* ; *être*}, {*dans des contextes* ; *lorsque*}, {*mauvais placement* ; *placé incorrectement*}, {*incidence* ; *complications possibles*}...

  Dans plusieurs cas, ces différents procédés de simplification lexicale (définitions, équivalents, substitutions) sont employés en même temps dans une même phrase ;

— *Simplification syntaxique.* Le fait le plus marquant de la simplification syntaxique concerne les énumérations et les exemples virgulés. Ainsi, une phrase coordonnée peut être segmentée en une liste avec des items. Cependant, il n'y a pas de règles sur ce qui est approprié à faire

pour effectuer la simplification car parfois les énumérations virgulées se trouvent dans les documents techniques et dans d'autres cas dans les documents simplifiés ;

— *Style.* Dans le corpus *Médicaments*, certains énoncés deviennent personnels et s'adressent directement à la personne grâce à l'emploi de pronoms personnels (*vous, votre, vos...*), comme dans les exemples de la figure 3.

5.3 Substitution lexicale

Pour l'évaluation de la substitution lexicale, nous avons sélectionné aléatoirement 10 documents dans chacun des trois corpus techniques pour y appliquer la méthode. Cela représente 7 892 phrases (2 456 pour le corpus *Médicaments*, 5 057 pour le corpus *Encyclopédie*, 379 pour le corpus *Cochrane*). La substitution lexicale, effectuée avec la ressource issue de Wiktionary, a permis de traiter 86 phrases. Cette faible couverture suggère que des ressources plus spécifiques sont nécessaires.

Critère	*% Méthode*	*Devlin*	*Biran*
Grammaticalité	70%	70.23%	77.91%
Simplicité	14.46%	46.43%	75.58%
Sémantique	18.51%	55.95%	46.43%

TABLE 5 – Évaluation manuelle des substitutions.

Dans le tableau 5, colonne *% Méthode*, nous indiquons les résultats d'évaluation des substitutions effectuées. Globalement, les substitutions fournissent des résultats qui restent grammaticaux : la ressource utilisée contient des séries de synonymes et d'hyperonymes qui appartiennent le plus souvent à la même catégorie grammaticale, comme {*absorption* ; *ingestion*} ou {*traiter* ; *soigner*}. Concernant la simplicité, les substitutions n'apportent pas toujours la simplification des phrases d'origine : un filtrage supplémentaire ou différent de la ressource est nécessaire. Finalement, la substitution peut aussi introduire des nuances sémantiques dans les phrases traitées. Nous comparons nos résultats avec deux travaux en substitution lexicales effectués en anglais (Devlin & Unthank, 2006; Biran *et al.*, 2011) : la ressource WordNet est exploitée pour traiter des textes de la langue générale. Nos résultats sont comparables quant à la grammaticalité, en revanche nous obtenons des résultats de simplicité et de sémantique plus faibles. Nous pensons que la raison principale de ces faibles résultats vient de la ressource utilisée, qui n'est pas adaptée à la simplification de textes médicaux techniques ou spécialisés. Des ressources plus spécifiques sont donc nécessaires.

Les figures 4 et 5 proposent quelques exemples de substitutions effectuées avec les ressources disponibles. Ainsi, la figure 4 propose des substitutions réussies, où la sémantique des phrases reste fidèle aux phrases d'origine, grâce aux synonymes comme {*absorption* ; *ingestion*}, {*traitement* ; *prescription*} ou {*traiter* ; *soigner*}. Alors que la figure 5 propose des substitutions non réussies, où la sémantique des phrases n'est pas sauvegardée. Par exemple, la sémantique change dans le cas des synonymes {*corps* ; *mort*}, alors que dans l'exemple avec les synonymes comme {*main* ; *pince*}, {*dents* ; *chicots*} ou {*tête* ; *citron*}, il s'agit de synonymes qui appartiennent à différents niveaux de la langue {*normé* ; *jargon*}. Même si cela ne modifie pas beaucoup la sémantique des phrases, la formulation devient plus familière, ce qui n'etait pas l'effet recherché.

Avant substitution	*Après substitution*
La nourriture n'a pas d'effet sur l'absorption d'anastrozole.	La nourriture n'a pas d'effet sur l'ingestion d'anastrozole.
Vous devez discuter avec votre médecin sur les risques et les options de traitement.	Vous devez discuter avec votre médecin sur les risques et les options de prescription.
Votre médecin peut vous prescrire un médicament visant à prévenir ou traiter cette perte osseuse.	Votre médecin peut vous prescrire un médicament visant à prévenir ou soigner cette perte osseuse.

FIGURE 4 – Exemples de substitutions réussies.

Avant substitution	*Après substitution*
Un abcès est une accumulation de pus sous la peau ou à l'intérieur du corps.	Un abcès est une accumulation de pus sous la peau ou à l'intérieur du mort.
Syndrome du canal carpien (fourmillement, douleur, sensation de froid, faiblesse dans certaines parties de la main).	Syndrome du canal carpien (fourmillement, douleur, sensation de froid, faiblesse dans certaines parties de la pince).

FIGURE 5 – Exemples de substitutions non réussies.

6 Conclusion et perspectives

Dans ce travail, nous avons proposé d'effectuer la simplification automatique de textes médicaux en français. Notre travail propose plusieurs contributions : (1) création de corpus comparables avec des textes médicaux techniques et simplifiés ; (2) alignement manuel de phrases ; (3) observations des procédés de simplification présents dans les corpus ; (4) premiers tests de substitution lexicale ; (5) évaluation des résultats avec trois critères de jugement (grammaticalité, simplification et sémantique).

Nous avons plusieurs perspectives à ce travail : (1) préparer et exploiter un lexique plus approprié pour la substitution lexicale dans les textes médicaux, comme ceux proposés dans les travaux existants (Grabar & Hamon, 2016), ce qui devrait permettre d'augmenter la couverture des substitutions ; (2) mieux gérer l'ambiguïté contextuelle des synonymes, ce qui devrait permettre d'augmenter l'acceptabilité sémantique des substitutions ; (3) augmenter le volume de phrases alignées, ce qui devrait permettre de tester d'autres approches pour la substitution, y compris les approches probabilistes ; (4) combiner différents types de modifications lexicales (substitutions, ajouts de paraphrases et de définitions) ; (5) combiner la simplification lexicale avec la simplification syntaxique pour fournir des résultats plus complets.

Remerciements

La présente publication s'inscrit dans le projet *CLEAR (Communication, Literacy, Education, Accessibility, Readability)* financé par l'ANR sous la référence ANR-17-CE19-0016-01.

Je remercie les relecteurs pour leurs remarques constructives. Je remercie également Natalia Grabar, pour son aide dans la réalisation des travaux décrits ici, ainsi que dans la rédaction de cette publication.

Références

ABUALHAIJA S., MILLER T., ECKLE-KOHLER J., GUREVYCH I. & ZIMMERMANN K.-H. (2017). Metaheuristic approaches to lexical substitution and simplification. In *EACL 2017*, p. 1–11.

AMOIA M. & ROMANELLI M. (2012). SB : mmSystem - using decompositional semantics for lexical simplification. In **SEM 2012*, p. 482–486, Montréal, Canada.

BARBU E., MARTIN-VALDIVIA M., ALFONSO L. & LOPEZ U. (2013). Open book : a tool for helping ASD users' semantic comprehension. In *Proceedings of the 2nd workshop of natural language processing for improving textual accessibility NLP4ITA*, p. 11–19, Atlanta, United States.

BERKMAN N., SHERIDAN S., DONAHUE K., HALPERN D. & CROTTY K. (2011). Low health literacy and health outcomes : An updated systematic review. *Annals of Internal Medicine*, **155**(2), 97–107.

BIRAN O., BRODY S. & ELHADAD N. (2011). Putting it simply : a context-aware approach to lexical simplification. In ACL, Ed., *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies : short papers-Volume 2*, p. 496–501.

BRIN-HENRY F. (2014). Éducation thérapeutique du patient et orthophonie. In *Communiquer malgré l'aphasie*. S. Médical.

BROUWERS L., BERNHARD D., LIGOZAT A.-L. & FRANÇOIS T. (2014). Syntactic Sentence Simplification for French. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR) @ EACL 2014*, p. 47–56, Gothenburg, Sweden.

CHANDRASEKAR R., DORAN C. & SRINIVAS B. (1996). Motivations and methods for text simplification. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, p. 1041–1044, Stroudsburg, PA, USA : Association for Computational Linguistics.

CHMIELIK J. & GRABAR N. (2011). *Détection de la spécialisation scientifique et technique des documents biomédicaux grâce aux informations morphologiques*, In *TAL*, volume 51(2), p. 151–179.

CÔTÉ R. (1996). *Répertoire d'anatomopathologie de la SNOMED internationale, v3.4*. Université de Sherbrooke, Sherbrooke, Québec.

DAVIS T. & WOLF M. (2004). Health literacy : implications for family medicine. *Fam Med*, **36**, 595–598.

DE BELDER J. & MOENS M. (2010). Text simplification for children. In *Workshop on accessible search systems of SIGIR*, p. 1–8.

DEVLIN S. & UNTHANK G. (2006). Helping aphasic people process online information. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '06, p. 225–226, New York, NY, USA : ACM.

DUBAY W. (2004). *The principles of readability*, In *Impact Information*.

FLESCH R. (1948). *A new readability yardstick*, In *Jounal of Applied Psychology*, volume 23, p. 221–233.

FRANÇOIS T., BILLAMI M. B., GALA N. & BERNHARD D. (2016). Automatic ranking of synonyms according to their reading and comprehension difficulty. In *JEP-TALN-RECITAL 2016*, volume 2 of *TALN*, p. 15–28, Paris, France.

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

GASPERIN C., MAZIERO E., SPECIA L., PARDO T. & ALUISIO R. M. (2009). *Natural language processing for social inclusion : a text simplification architecture for different literacy levels*, In *SEMISH-XXXXVI*, p. 397–404.

GLAVAS G. & STAJNER S. (2015). Simplifying lexical simplification : Do we need simplified corpora ? In *ACL-COLING*, p. 63–68.

GOEURIOT L., GRABAR N. & DAILLE B. (2007). *Caractérisation des discours scientifique et vulgarisé en français, japonais et russe*, In *TALN*, p. 93–102.

GRABAR N. & HAMON T. (2016). Exploitation de la morphologie pour l'extraction automatique de paraphrases grand public des termes médicaux. *TAL*, **57**(1), 85–109.

GUNNING R. (1973). *The art of clear writing*. New York, NY : McGraw Hill.

HORN C., MANDUCA C. & KAUCHAK D. (2014). Learning a lexical simplifier using Wikipedia. In *ACL Annual Meeting*, p. 458–463.

INUI K., FUJITA A., TAKAHASHI T., IIDA R. & IWAKURA T. (2003). Text simplification for reading assistance : a project note. In *Proc. of the 2nd international workshop on paraphrasing : paraphrase acquisition and applications*, p. 9–16.

JAUHAR S. & SPECIA L. (2012). UOW-SHEF : SimpLex – lexical simplicity ranking based on contextual and psycholinguistic features. In **SEM 2012*, p. 477–481, Montréal, Canada.

JOHANNSEN A., MARTÍNEZ H., KLERKE S. & SØGAARD A. (2012). Emnlp@cph : Is frequency all there is to simplicity ? In **SEM 2012*, p. 408–412, Montréal, Canada.

JONNALAGADDA S., TARI L., HAKENBERG J., BARAL C. & GONZALEZ G. (2009). Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume : Short Papers*, p. 177–180 : Association for Computational Linguistics.

KOKKINAKIS D. & GRONOSTAJ M. T. (2006). *Comparing lay and professional language in cardiovascual disordes corpora*, In *WSEAS transactions on biology and biomedicine*, p. 429–437.

LIGOZAT A., GROUIN C., GARCIA-FERNANDEZ A. & BERNHARD D. (2012). Annlor : A naïve notation-system for lexical outputs ranking. In **SEM 2012*, p. 487–492.

POPRAT M., MARKÓ K. & HAHN U. (2006). *A language classifier that automatically divides medical documents for experts and health care consumers*, In *MIE 2006 – Proceedings of the XX international congress of the european federation for medical informatics*, p. 503–508. Maastricht.

RELLO L., BAEZA-YATES R. A., BOTT S. & SAGGION H. (2013). Simplify or help ? : text simplification strategies for people with dyslexia. In *W4A*.

SACKETT D. L., ROSENBERG W. M. C., GRAY J. A. M., HAYNES R. B. & RICHARDSON W. S. (1996). Evidence based medicine : what it is and what it isn't. *BMJ*, **312**(7023), 71–72.

SAGGION H. (2017). *Automatic Text Simplification*. Morgan & Claypool Publishers.

SAJOUS F. & HATHOUT N. (2015). GLAWI, a free XML-encoded Machine-Readable Dictionary built from the French Wiktionary. In *Proceedings of the eLex 2015 conference*, p. 405–426, Herstmonceux, England.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees.

SERETAN V. (2012). *Acquisition of Syntactic Simplification Rules for French*. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA). ID : unige :30961.

SHARDLOW M. (2014). Out in the open : Finding and categorising errors in the lexical simplification pipeline. In N. CALZOLARI, K. COUKRI, T. DECLERCK, H. LOFTSSON, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Eds., *Proc. of the 9th International Conference on Language Resources and Evaluation*, LREC, Reykjavik, Iceland : European Language Resources Association (ELRA).

SINHA R. (2012). Unt-simprank : Systems for lexical simplification ranking. In **SEM 2012*, p. 493–496.

SPECIA L. (2010). *Translating from complex to simplified sentences*, In *International conference on computational processing of the portuguese language (Propor-2010)*, p. 30–39.

SPECIA L., JAUHAR S. & MIHALCEA R. (2012). Semeval-2012 task 1 : English lexical simplification. In **SEM 2012*, p. 347–355.

WILLIAMS S. & REITER E. (2005). Generating readable texts for readers with low basic skills. In *ENLG*.

WOODSEND K. & LAPATA M. (2011). *Learning to simplify sentences with quasi-synchronous grammar and integer programming*, In *EMNLP*, p. 409–420.

WUBBEN S., VAN DEN BOSCH A. & KRAHMER E. (2012). Sentence simplification by monolingual machine translation. In *ACL*, p. 1015–1024.

ZHU Z., BERNHARD D. & GUREVYCH I. (2010). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, p. 1353–1361, Stroudsburg, PA, USA : Association for Computational Linguistics.

Approche lexicale de la simplification automatique de textes médicaux

Rémi Cardon
UMR 8163 – STL – Savoirs Textes Langage, F-59000 Lille, France
remi.cardon@etu.univ-lille3.fr

RÉSUMÉ
Notre travail traite de la simplification automatique de textes. Ce type d'application vise à rendre des contenus difficiles à comprendre plus lisibles. À partir de trois corpus comparables du domaine médical, d'un lexique existant et d'une terminologie du domaine, nous procédons à des analyses et à des modifications en vue de la simplification lexicale de textes médicaux. L'alignement manuel des phrases provenant de ces corpus comparables fournit des données de référence et permet d'analyser les procédés de simplification mis en place. La substitution lexicale avec la ressource existante permet d'effectuer de premiers tests de simplification lexicale et indique que des ressources plus spécifiques sont nécessaires pour traiter les textes médicaux. L'évaluation des substitutions est effectuée avec trois critères : grammaticalité, simplification et sémantique. Elle indique que la grammaticalité est plutôt bien sauvegardée, alors que la sémantique et la simplicité sont plus difficiles à gérer lors des substitutions avec ce type de méthodes.

ABSTRACT
Lexical approach for the automatic simplification of medical texts

Our work addresses the automatic text simplification. This kind of application aims at improving the readability of texts that are difficult to read. Using three different corpora – which contain biomedical texts – an existing lexicon and a domain terminology, we perform analysis and modification of texts in order to achieve their lexical simplification. Manual alignment of sentences from comparable corpora provides reference data and permits to analyze the simplification procedures involved. Lexical substitution using existing resources permits to perform first tests of lexical simplification and indicates that specific resources are necessary when working with medical contents. The evaluation of substitutions is performed through three criteria : grammaticality, simplicity and semantics. It indicates that grammaticality is rather well preserved, while semantics and simplicity are more difficult to handle during the substitutions with this kind of methods.

MOTS-CLÉS : Simplification automatique de textes, analyse lexicale, domaine médical, simplification lexicale, substitution lexicale.

KEYWORDS: Automatic text simplification, lexical analysis, medical area, lexical simplification, lexical substitution.

1 Introduction

La simplification automatique de textes est un domaine du TAL, dans lequel il s'agit d'appliquer des transformations sur les phrases d'un texte pour les rendre plus lisibles, tout en conservant leur sens

intact. Cela est pratiqué aussi bien à destination des humains que pour faciliter les tâches nécessitant l'analyse automatique de textes (Chandrasekar *et al.*, 1996). La simplification, dont l'objectif est de faciliter des traitements d'analyse automatique, peut faire partie de différentes applications. Ainsi, la première application à l'avoir exploitée cherchait à simplifier les structures de phrases avant de procéder à leur analyse syntaxique automatique (Chandrasekar *et al.*, 1996). Dans d'autres contextes, la simplification peut être utilisée pour adapter certains genres de textes à des outils, qui n'ont pas été entraînés pour les traiter spécifiquement, comme par exemple l'analyse d'un texte biomédical effectuée avec des outils entraînés sur des textes journalistiques (Jonnalagadda *et al.*, 2009). Pour la simplification à destination des humains, ces méthodes sont explorées pour différents objectifs et différents publics. Notons par exemple que la simplification est effectuée pour des personnes avec une faible compétence de lecture (Williams & Reiter, 2005), pour des personnes sourdes qui montrent des difficultés de lecture et d'écriture (Inui *et al.*, 2003), pour des lecteurs dyslexiques (Rello *et al.*, 2013) ou encore pour des personnes autistes (Barbu *et al.*, 2013). Dans le domaine médical – dans lequel nous nous plaçons ici – la simplification peut également servir à faciliter l'éducation thérapeutique des patients (Brin-Henry, 2014) ou l'accès à l'information par les enfants (De Belder & Moens, 2010). En effet, des études ont montré qu'une meilleure compréhension des informations de santé par les patients et leurs familles mène à une meilleure adhésion au traitement et à un processus de soins plus réussi (Davis & Wolf, 2004; Berkman *et al.*, 2011).

L'objectif de notre travail consiste à contribuer au domaine de la simplification de textes de spécialité, sur l'exemple de textes médicaux. D'une part, nous proposons de constituer des corpus comparables différenciés par leur spécialisation, d'effectuer un alignement de phrases à partir de ces corpus afin de faire une analyse des procédés de simplification mis en place. D'autre part, nous proposons d'effectuer de premiers tests de simplification lexicale en utilisant la substitution. Si la majorité de travaux de simplification traitent les données en langue anglaise, nous travaillons avec les données en français.

Dans la suite de ce travail, nous présentons d'abord l'état de l'art (section 2), ensuite les données sur lesquelles nous travaillons (section 3). La méthode est présentée dans la section 4 et les résultats dans la section 5. Nous proposons une conclusion et les perspectives de ce travail dans la section 6.

2 État de l'art

Les travaux en simplification automatique se positionnent essentiellement à deux niveaux : lexical et syntaxique. La *simplification lexicale* opère au niveau des unités lexicales. Nous allons illustrer la simplification lexicale avec la tâche de simplification proposée lors de la compétition *SemEval 2012* [1] pour la langue anglaise. Pour un texte court et un mot cible, plusieurs substitutions possibles et satisfaisant le contexte ont été proposées par les organisateurs. L'objectif consistait à trier ces substitutions selon leur degré de simplicité (Specia *et al.*, 2012) et donc de les positionner les unes par rapport aux autres, en fonction de leur difficulté. Par exemple, pour la phrase *Hitler committed terrible atrocities during the second World War*, le mot à substituer est *atrocities*. Les candidats synonymes proposés par les organisateurs sont *abomination, cruelty, enormity, violation*. Le choix de référence est *cruelty*, ce qui doit produire en sortie : *Hitler committed terrible cruelties during the second World War.* De manière générale, lors de la simplification lexicale, plusieurs étapes peuvent être distinguées :

1. L'identification de mots ou termes qui peuvent poser des difficultés de compréhension. Cette étape est le plus souvent accomplie à l'aide de ressources lexicales auxquelles sont associées

1. http://www.cs.york.ac.uk/semeval-2012/

des mesures de complexité des mots, même si ces mesures n'ont pas reçu le consensus de la communauté de recherche (Saggion, 2017; Shardlow, 2014). Comme indiqué plus bas, des mesures classiques et computationnelles, et donc plus récentes, sont distinguées ;

2. Le remplacement de ces unités par un équivalent jugé plus facile de compréhension. Cette étape repose sur la disponibilité d'un dictionnaire d'expressions synonymiques (Shardlow, 2014) ou même hyperonymiques ;
3. Lorsque plusieurs équivalents sont disponibles, il est nécessaire de les ordonner par rapport à leur niveau de difficulté pour être en mesure de sélectionner les candidats les plus faciles à comprendre (François *et al.*, 2016). C'était typiquement la tâche proposée lors de la compétition *SemEval 2012*. Les participants ont exploité plusieurs critères pour effectuer cette tâche : lexique d'un corpus oral et de Wikipedia, n-grammes de Google, WordNet (Sinha, 2012) ; longueur de mots, nombre des syllabes, information mutuelle, fréquences (Jauhar & Specia, 2012) ; fréquences dans Wikipedia, longueur de mots, n-grammes, complexité syntaxique des documents (Johannsen *et al.*, 2012) ; n-grammes, fréquences dans Wikipedia, n-grammes de Google (Ligozat *et al.*, 2012) ; WordNet, fréquences (Amoia & Romanelli, 2012) ;
4. Il faut également s'assurer que les candidats à substitution sont acceptables dans le contexte de chaque phrase traitée. Dans *SemEval 2012*, cette condition était assurée par les organisateurs.

La *simplification syntaxique* opère au niveau de la syntaxe. Son objectif est de réorganiser la structure syntaxique des phrases. Quelques exemples d'opérations de réorganisation de cette structure sont : le découpage de phrases complexes en plusieurs phrases plus simples, l'ajout ou la suppression de propositions, la modification de temps verbaux (Brouwers *et al.*, 2014; Gasperin *et al.*, 2009; Seretan, 2012). Pour un exemple, la phrase *Elle compte, selon les autorités du pays, en 2009, 5 878 609 habitants pour l'agglomération, et 3 796 677 habitants pour la ville, soit 20 % de la population totale du pays* devient, après l'application d'une règle de suppression de propositions subordonnées et d'incises : *Elle compte, selon les autorités du pays, en 2009, 5 878 609 habitants pour l'agglomération, et 3 796 677 habitants pour la ville* (Brouwers *et al.*, 2014).

Pour ces deux types de simplification, des approches à base de règles et de probabilités ont été développées. Les approches à base de règles reposent sur l'expertise des concepteurs et leur connaissance des procédés de simplification. Notons que cette expertise peut également profiter d'un corpus de simplification déjà disponible. Les approches statistiques nécessitent de disposer de corpus de textes comparables, ou même parallèles et alignés, qui contiennent typiquement des textes complexes et leurs versions simplifiées (Zhu *et al.*, 2010; Specia, 2010; Woodsend & Lapata, 2011). Un des exemples typiques de corpus comparables, qui sont largement utilisés dans ce type de travaux, sont Wikipedia [2] et Simple Wikipedia [3] en langue anglaise. Si le premier propose des articles à destination de la population en général, Simple Wikipedia vise des populations spécifiques (comme par exemple les enfants, les apprenants d'anglais, les adultes en difficulté de lecture, etc.).

Une des tâches importantes de la simplification automatique consiste à pouvoir mesurer la lisibilité des unités linguistiques. Différents niveaux de mesure sont utilisés, que ce soit au niveau des termes ou des textes. Deux grands types de mesures de lisibilité peuvent être distingués : classiques et computationnelles (François, 2011). Les mesures classiques reposent sur le calcul de la complexité de surface des mots (nombre de syllabes) et de phrases pour évaluer leur lisibilité. Par exemple, si les mots et les phrases d'un texte sont longs, il est considéré être difficile à lire (Flesch, 1948; Gunning, 1973; Dubay, 2004). Les mesures computationnelles fournissent la possibilité d'associer les

2. https://en.wikipedia.org/wiki/Main_Page
3. https://simple.wikipedia.org/wiki/Simple_English_Wikipedia

unités, dont on souhaite mesurer la lisibilité (mots, phrases, textes...), à une variété de descripteurs : combinaisons de mesures classiques avec des informations terminologiques (Kokkinakis & Gronostaj, 2006), utilisation de *n-grammes* de caractères (Poprat *et al.*, 2006), descripteurs discursifs (Goeuriot *et al.*, 2007), descripteurs morphologiques (Chmielik & Grabar, 2011), etc. En général, ces mesures sont calculées de manière supervisée par rapport à une référence et fournissent des résultats fiables.

Par rapport aux travaux existants, nous nous positionnons au niveau de la simplification lexicale. Nous proposons d'effectuer la simplification en effectuant des substitutions lexicales grâce à l'exploitation d'un lexique existant. Notre tâche est assez proche de celle proposée lors de *SemEval 2012*.

3 Présentation des données

Les données exploitées sont de deux types : les corpus comparables provenant du domaine médical, et les ressources utilisées pour l'analyse et la substitution lexicale.

3.1 Corpus

3.1.1 Cochrane

Cochrane [4] est un organisme qui a pour objectif la diffusion de l'information médicale (Sackett *et al.*, 1996). Les textes publiés par Cochrane sont des synthèses de la littérature médicale sur une question spécifique (diagnostic, traitement). Ces synthèses sont créées à destination des professionnels de santé. Plus récemment, elles sont simplifiées par les collaborateurs de Cochrane pour les rendre accessibles au grand public également. De même, les synthèses écrites en anglais sont traduites en d'autres langues, y compris le français. Le corpus que nous utilisons est composé de 3 815 synthèses en français à destination des médecins et, pour chacune d'elle, sa version simplifiée (voir la figure 1).

Version technique	*Version simplifiée*
L'hématome aigu de l'oreille est une affection qui se caractérise par la formation d'une collection sanguine sous le périchondre du pavillon. Il est souvent provoqué par un traumatisme contondant. En l'absence de traitement, il finit par entraîner une difformité couramment appelée oreille en chou-fleur ou oreille du boxeur.	L'hématome aigu de l'oreille est une affection qui se caractérise par la formation d'une collection sanguine dans le pavillon (oreille externe), souvent à la suite d'un traumatisme contondant. S'il n'est pas traité, il entraîne une difformité appelée oreille en chou-fleur ou oreille du boxeur.

FIGURE 1 – Exemple de textes comparables du corpus Cochrane.

3.1.2 Encyclopédie

Wikipedia [5] est une encyclopédie collaborative en ligne. Elle propose les informations à destination d'un large public et aborde un grand nombre de sujets. Vikidia [6] est également une encyclopédie

4. https://www.cochranelibrary.com
5. https://fr.wikipedia.org
6. https://fr.vikidia.org/

collaborative en ligne, sa spécificité est qu'elle est destinée aux enfants de 8 à 13 ans. Nous exploitons ces deux sources pour constituer un deuxième corpus de travail. Il est composé de 575 articles relatifs au portail de la Médecine de Wikipedia et des articles équivalents de Vikidia (voir la figure 2).

Version technique	*Version simplifiée*
La luette ou uvule est un appendice conique situé au fond de la cavité buccale et proche des tonsilles palatines. Le mot uvule vient du latin uva, qui signifie "grain de raisin". La luette est un organe de 10 à 15 millimètres de long, de forme tubulaire quand il est détendu, qui pend à la partie moyenne du bord inférieur du voile du palais. Elle est constituée d'un tissu membraneux et musculaire.	La luette ou uvule est un appendice conique située au fond de la bouche. C'est un organe fait de tissus membraneux et musculaires, d'environ 10 à 15mm de long, qui pend à la partie moyenne du voile du palais.

FIGURE 2 – Exemple de textes comparables du corpus Wikipedia/Vikidia.

3.1.3 Médicaments

Le troisième corpus contient les informations sur les médicaments issues de la base de données [7] du ministère de la santé. On peut accéder, pour un médicament donné, au résumé des caractéristiques du produit (RCP), créé à destination des professionnels, et à la notice, créée à destination du grand public. Ces dernières peuvent aussi être trouvées dans les boîtes de médicaments. Ce corpus contient 11 800 RCP techniques et leurs notices grand public (voir la figure 3).

Version technique	*Version simplifiée*
- hypersensibilité à l'huile de paraffine	- si vous avez une allergie à l'huile de paraffine
- colopathie obstructive, compte tenu de l'effet laxatif du médicament	- si vous êtes atteint de colopathie obstructive, compte tenu de l'effet laxatif du médicament
- syndrome douloureux abdominal de cause indéterminée et inflammatoire (rectocolite ulcéreuse, maladie de Crohn)	- si vous êtes atteint de syndrome douloureux abdominal de cause indéterminée et inflammatoire (rectocolite ulcéreuse, maladie de Crohn, ...)
- ne pas utiliser chez les personnes présentant des difficultés de déglutition en raison du risque d'inhalation bronchique et de pneumopathie lipoïde	- ne pas utiliser chez les personnes présentant des difficultés pour avaler en raison du risque d'inhalation de la paraffine liquide qui entraîne une pneumopathie lipoïde

FIGURE 3 – Exemple de textes comparables du corpus Médicament.

3.1.4 Bilan des corpus

Le tableau 1 fait le bilan des trois corpus et indique, pour chaque corpus : sa taille en nombre de documents, d'occurrences de mots et de lemmes uniques. Il s'agit de corpus comparables : les articles

7. https://base-donnees-publique.medicaments.gouv.fr/

concernent les mêmes sujets mais relèvent de différents discours. Comme le montrent les extraits, ils proposent rarement une réécriture de la version technique (originale) en langage simplifié. En effet, le plus souvent, la création de la version simplifiée semble être indépendante de la version technique. Nous pouvons voir que le corpus *Médicaments* est le plus gros des trois corpus étudiés, tandis que le corpus *Encyclopédie* est le plus petit. Par ailleurs, les versions techniques sont toujours plus volumineuses que les versions simplifiées.

Corpus	*nb doc.*	*nb occ.*	*nb lemmes uniques*
Cochrane technique	3 815	2 804 336	11 558
Cochrane simplifié	3 815	1 491 243	7 567
Médicaments technique	11 800	51 705 111	43 515
Médicaments simplifié	11 800	33 116 119	25 725
Wikipedia	575	2 186 891	19 287
Vikidia	575	183 051	3 117

TABLE 1 – Taille des corpus comparables.

3.2 Ressources

Nous utilisons deux ressources : une terminologie spécialisée, Snomed International, et un lexique généraliste issu du Wiktionary.

3.2.1 Terminologie médicale Snomed

Nous exploitons la terminologie médicale Snomed International (Côté, 1996) telle que diffusée par ASIP santé. [8] La vocation de cette terminologie est de décrire le domaine médical. La terminologie contient 151 104 termes médicaux structurés en onze axes sémantiques (maladies et anomalies, actes médicaux, produits chimiques, organismes vivants, anatomie). Selon notre hypothèse, le contenu de cette terminologie permet d'estimer la couverture en termes et mots médicaux d'un texte donné.

3.2.2 Lexique issu du Wiktionary

Nous utilisons un lexique obtenu à partir des articles du Wiktionary [9], dans sa version GLAWI (Sajous & Hathout, 2015). Nous retenons les entrées dont au moins une définition est associée à l'une des catégories suivantes : *anat*, *chirurgie*, *génétique*, *maladie(s)*, *médecine*, *médicaments*, *microbiologie*, *neurologie*, *pathologie*, *pédologie*, *pharmacologie*, *physiologie*, *squelette*, *virologie*. Cela fournit un lexique avec 8 012 entrées uniques qui peuvent être liées au domaine médical. Les catégories les plus importantes sont *médecine* avec 4 967 entrées et *anat* (pour *anatomie*) avec 1 925 entrées. Les autres catégories contiennent entre quelques dizaines et quelques centaines d'entrées.

Cette ressource fournit des séries de synonymes et des hyperonymes pour certains termes. Ainsi, 25 % des entrées de GLAWI ont au moins un synonyme, avec une moyenne de 2,42 synonymes par entrée.

8. http://esante.gouv.fr/services/referentiels/referentiels-d-interoperabilite/snomed-35vf

9. https://fr.wiktionary.org/

Seules 452 entrées ont des hyperonymes : 318 avec un hyperonyme et 134 avec 2 à 6 hyperonymes. Les séries de synonymes et d'hyperonymes comportent le plus souvent des mots de la même partie du discours. Cette ressource contient essentiellement les entrées simples, mais nous avons également 1 157 entrées polylexicales. Enfin, cette ressource offre le paradigme flexionnel (le lemme et ses formes fléchies) des entrées.

4 Méthode

4.1 Pré-traitements

Les corpus sont pré-traités : l'étiquetage morpho-syntaxique et la lemmatisation sont effectués avec le TreeTagger (Schmid, 1994). Cela permet de normaliser le contenu des corpus grâce à la lemmatisation. Les termes de la terminologie Snomed International sont projetés sur les textes, ce qui permet d'effectuer une analyse lexicale de ces corpus.

4.2 Alignement de phrases

Nos corpus sont des corpus comparables. Notre objectif est de constituer une base de phrases alignées, qu'elles soient parallèles ou comparables, à partir desquelles nous pourrions faire des observations sur les manières dont la simplification peut être effectuée.

Une sélection aléatoire d'articles a fourni : 2*13 documents du corpus *Cochrane*, 2*12 documents du corpus *Médicaments* et 2*14 documents du corpus *Encyclopédie*. Ces articles sont utilisés pour effectuer l'alignement manuel de phrases provenant des corpus techniques et simplifiés. Deux annotateurs ont effectué l'alignement de manière indépendante. Des séances de consensus ont permis ensuite de résoudre les désaccords. Les critères qui guidaient l'alignement sont les suivants :

1. Les deux phrases doivent avoir le même sens ou sinon des sens proches, comme dans ces phrases du corpus *Cochrane* :
   — *les sondes gastriques sont couramment utilisées pour administrer des médicaments ou une alimentation entérale aux personnes ne pouvant plus avaler*
   — *les sondes gastrique sont couramment utilisées pour administrer des médicaments et de la nourriture directement dans le tractus gastro-intestinal (un tube permettant de digérer les aliments) pour les personnes ne pouvant pas avaler*
2. Le sens d'une phrase peut se retrouver intégralement dans le sens de l'autre phrase. Dans l'idéal, il devrait s'agir de l'inclusion sémantique, comme dans cet exemple, où la phrase technique indique le nombre de participants et la mesure d'évaluation en plus :
   — *peu de données (43 participants) étaient disponibles concernant la détection d'un mauvais placement (la spécificité) en raison de la faible incidence des mauvais placements*
   — *cependant, peu de données étaient disponibles concernant les sondes placées incorrectement et les complications possibles d'une sonde mal placée*
3. Les cas d'intersection sémantique, où chaque phrase apporte des informations spécifiques, seraient à proscrire. L'exemple du corpus *Cochrane* qui suit illustre ce cas :
   — *des études à plus grande échelle sont nécessaires pour déterminer la possibilité d'événements indésirables lorsque les ultrasons sont utilisés pour confirmer le positionnement des sondes*

— *des études à plus grande échelle sont nécessaires pour déterminer si les ultrasons pourraient remplacer les rayons x pour confirmer la mise en place d'une sonde gastrique, et pour évaluer si les ultrasons pourraient permettre de réduire les complications graves, telles que la pneumonie résultant d'un tube mal placé*

Les cas d'intersection sémantique sont plus difficiles à généraliser et à reproduire.

Dans ces exemples, nous voyons que le passage d'une phrase technique vers sa version simplifiée requiert des modifications syntaxiques et lexicales. Dans notre travail, nous nous concentrons sur la simplification lexicale effectuée au moyen de substitutions lexicales.

4.3 Substitution lexicale

Nous proposons d'aborder la simplification au niveau lexical grâce aux substitutions de mots par leurs équivalents supposés être plus simples et faciles à comprendre. Notre approche est fondée sur des règles. Elle suit les étapes suivantes :

— Les phrases des corpus techniques sont exploitées pour effectuer les tests de substitution ;

— L'ensemble médical du lexique Wiktionary est exploité pour fournir les candidats à substitution. Ce lexique propose en effet des ensembles de synonymes et d'hyperonymes ;

— Les entrées de ce lexique sont filtrées. Comme nous l'avons vu, le corpus *Vikidia* propose le contenu le moins spécialisé par rapport au reste des corpus. Nous projetons donc les entrées du lexique et leurs synonymes et hyperonymes sur le corpus *Vikidia*. Si une entrée, ses synonymes ou hyperonymes, sont reconnues dans ce corpus, nous supposons qu'il s'agit des entrées plus faciles à comprendre, et les retenons pour effectuer les substitutions lexicales ;

— Si une phrase contient une entrée du lexique qui possède des synonymes ou hyperonymes, si cette entrée ne se trouve pas dans le corpus *Vikidia*, et si un de ses synonymes ou hyperonymes se trouve dans le corpus *Vikidia*, ce synonyme ou hyperonyme est utilisé pour la substitution.

Il s'agit d'une méthode souvent exploitée dans les travaux de l'état de l'art, qui visent à effectuer la simplification lexicale des textes (Biran *et al.*, 2011; Wubben *et al.*, 2012; Horn *et al.*, 2014; Glavas & Stajner, 2015; Abualhaija *et al.*, 2017).

4.4 Évaluation

Les résultats de la substitution lexicale sont évalués avec plusieurs critères exploités dans les travaux existants en simplification lexicale (Biran *et al.*, 2011; Wubben *et al.*, 2012) :

— *Grammaticalité*. Le jugement sur la grammaticalité doit répondre à la question de savoir si la phrase reste grammaticale après les modifications effectuées. Pour assurer le respect de ce critère, lors de la simplification lexicale par exemple, la plupart des travaux effectuent la substitution avec des mots de la même catégorie syntaxique que le mot substitué (Biran *et al.*, 2011; Horn *et al.*, 2014; Glavas & Stajner, 2015; Abualhaija *et al.*, 2017) ;

— *Sémantique*. Le jugement sur la sémantique doit répondre à la question de savoir si la transformation effectuée préserve la sémantique originale de la phrase. En effet, quelle que soit la simplification effectuée, la sémantique des textes doit rester préservée ;

— *Simplicité*. Le jugement sur la simplicité doit répondre à la question de savoir si la simplification effectuée sur une phrase la rend plus simple à comprendre.

Ces critères sont évalués manuellement par l'auteur.

5 Résultats et leur discussion

Nous présentons les résultats relatifs à l'analyse lexicale des corpus (section 5.1), à l'alignement de phrases (5.2) et à la substitution lexicale en vue de simplification (section 5.3).

5.1 Analyse lexicale

L'analyse lexicale, basée sur la terminologie Snomed International, permet de calculer : (1) le nombre de ses mots et termes qui apparaissent dans chacun des corpus, et (2) le ratio d'occurrences des termes de la Snomed entre les textes en version technique et simplifiée. Notons que la lemmatisation avec TreeTagger peut empêcher la reconnaissance d'expressions polylexicales présentes dans la terminologie, comme {*bilirubine totale* ; *bilirubine total*} ou {*amnésie passagère* ; *amnésie passager*}. L'objectif de la projection ici est d'obtenir une estimation du degré de spécialisation de chaque sous-corpus, et plus précisément de pouvoir comparer ces estimations entre elles. Nous n'appliquons pas de traitement spécifique pour le repérage des expressions polylexicales.

Corpus	*Termes*
Cochrane simplifié	2 316
Cochrane technique	2 505
Médicaments simplifié	2 700
Médicaments technique	3 332
Encyclopédie simplifié	1 635
Encyclopédie technique	3 999

TABLE 2 – Nombre de termes uniques de la Snomed International dans les corpus.

Ratio	*Valeur*
Cochrane : simplifié / technique	0.60
Cochrane : technique / simplifié	1.66
Médicaments : simplifié / technique	0.62
Médicaments : technique / simplifié	1.62
Encyclopédie : simplifié / technique	0.10
Encyclopédie : technique / simplifié	9.67

TABLE 3 – Ratio des occurrences de termes de la Snomed International dans les corpus.

Les tableaux 2 et 3 présentent les résultats. Selon le tableau 2, nous trouvons plus de termes de la Snomed dans les versions techniques. Par ailleurs, le corpus Vikidia est le plus pauvre en termes spécialisés. Il s'agit certainement du corpus dont le niveau de lisibilité est le plus élevé. Ces observations sont corroborées par les indications du tableau 3 qui indique les ratios de termes entre les corpus techniques et spécialisés. Ces ratios sont comparables pour les corpus *Cochrane* et *Médicaments*. En revanche, nous observons une différence beaucoup plus importante entre les versions technique et simplifiée du corpus encyclopédique, où les versions simplifiées d'articles contiennent beaucoup moins de termes spécialisés.

5.2 Alignement de phrases

Corpus	*nb doc.*	*nb phrases total*	*nb phrases alignées*	*ratio*
Cochrane	26	653	240	36,75%
Médicaments	24	7101	258	3,63%
Encyclopédie	28	2651	163	6,15%

TABLE 4 – Nombre de phrases alignées pour chaque corpus.

Le tableau 4 indique les résultats consensuels de l'alignement manuel des phrases. Nous pouvons observer que les phrases alignées, qui font la correspondance entre le contenu technique et simplifié, sont relativement plus rares pour les corpus *Médicaments* et *Encylopédie*, alors que le corpus *Cochrane* en offre plus par rapport à sa taille. Les raisons de cet état de chose peuvent être les suivantes :

— La ligne directrice de rédaction des versions simplifiées des résumés de la fondation Cochrane affiche explicitement une volonté de simplifier le contenu de ses résumés d'origine pour le grand public. Les rédacteurs et traducteurs prennent donc comme point de départ les résumés originaux et techniques et les simplifient au fur et à mesure de l'avancement ;

— Pour les deux autres corpus, les principes ne sont pas aussi stricts. Ainsi, l'objectif de Vikidia est de traiter des sujets présents dans Wikipedia mais pour un public d'enfants. La création d'articles de Vikidia est rarement basée sur les articles de Wikipedia : le plus souvent, il s'agit d'une écriture indépendante. Quant au corpus *Médicaments*, les mêmes médicaments sont décrits et spécifiés dans les versions technique et simplifiée. Cependant, certaines informations sur les médicaments sont propres aux RCP (composition plus détaillée, action sur l'organisme, molécules, détail sur les effets indésirables...), alors que d'autres informations sont propres aux notices destinées au grand public (précautions d'emploi, mises en garde...).

Une analyse de ces phrases alignées nous indique aussi que les procédés de simplification (lexicale, syntaxique et stylistique) ne sont pas les mêmes selon les corpus :

— *Simplification lexicale.* Dans Vikidia, les notions complexes sont plutôt explicitées, alors que dans les corpus *Médicaments* et *Cochrane*, les notions complexes sont souvent suivies par leurs équivalents entre parenthèses :

  — *en revanche, les ultrasons associés à d'autres tests (par exemple, la visualisation de l'irrigation saline (injecter une solution saline à travers la sonde et l'observer à l'intérieur de l'estomac par ultrasons)) pourraient être utiles pour confirmer le placement des tubes utilisés pour le drainage gastrique*

  — *l'alimentation offerte au travers d'un tube placé par erreur dans la trachée (un conduit où passe l'air respiré) peut entraîner une pneumonie grave (une infection des poumons)*

  Quel que soit le corpus, les notions complexes peuvent aussi être remplacées par leurs équivalents plus simples. En voilà quelques exemples au format {*technique* ; *simplifié*} :

  {*alimentation* ; *nourriture*}, {*entérale* ; *directement dans le tractus gastro-intestinal*}, {*fournir* ; *être*}, {*dans des contextes* ; *lorsque*}, {*mauvais placement* ; *placé incorrectement*}, {*incidence* ; *complications possibles*}...

  Dans plusieurs cas, ces différents procédés de simplification lexicale (définitions, équivalents, substitutions) sont employés en même temps dans une même phrase ;

— *Simplification syntaxique.* Le fait le plus marquant de la simplification syntaxique concerne les énumérations et les exemples virgulés. Ainsi, une phrase coordonnée peut être segmentée en une liste avec des items. Cependant, il n'y a pas de règles sur ce qui est approprié à faire

pour effectuer la simplification car parfois les énumérations virgulées se trouvent dans les documents techniques et dans d'autres cas dans les documents simplifiés ;

— *Style.* Dans le corpus *Médicaments*, certains énoncés deviennent personnels et s'adressent directement à la personne grâce à l'emploi de pronoms personnels (*vous, votre, vos...*), comme dans les exemples de la figure 3.

5.3 Substitution lexicale

Pour l'évaluation de la substitution lexicale, nous avons sélectionné aléatoirement 10 documents dans chacun des trois corpus techniques pour y appliquer la méthode. Cela représente 7 892 phrases (2 456 pour le corpus *Médicaments*, 5 057 pour le corpus *Encyclopédie*, 379 pour le corpus *Cochrane*). La substitution lexicale, effectuée avec la ressource issue de Wiktionary, a permis de traiter 86 phrases. Cette faible couverture suggère que des ressources plus spécifiques sont nécessaires.

Critère	*% Méthode*	*Devlin*	*Biran*
Grammaticalité	70%	70.23%	77.91%
Simplicité	14.46%	46.43%	75.58%
Sémantique	18.51%	55.95%	46.43%

TABLE 5 – Évaluation manuelle des substitutions.

Dans le tableau 5, colonne *% Méthode*, nous indiquons les résultats d'évaluation des substitutions effectuées. Globalement, les substitutions fournissent des résultats qui restent grammaticaux : la ressource utilisée contient des séries de synonymes et d'hyperonymes qui appartiennent le plus souvent à la même catégorie grammaticale, comme {*absorption* ; *ingestion*} ou {*traiter* ; *soigner*}. Concernant la simplicité, les substitutions n'apportent pas toujours la simplification des phrases d'origine : un filtrage supplémentaire ou différent de la ressource est nécessaire. Finalement, la substitution peut aussi introduire des nuances sémantiques dans les phrases traitées. Nous comparons nos résultats avec deux travaux en substitution lexicales effectués en anglais (Devlin & Unthank, 2006; Biran *et al.*, 2011) : la ressource WordNet est exploitée pour traiter des textes de la langue générale. Nos résultats sont comparables quant à la grammaticalité, en revanche nous obtenons des résultats de simplicité et de sémantique plus faibles. Nous pensons que la raison principale de ces faibles résultats vient de la ressource utilisée, qui n'est pas adaptée à la simplification de textes médicaux techniques ou spécialisés. Des ressources plus spécifiques sont donc nécessaires.

Les figures 4 et 5 proposent quelques exemples de substitutions effectuées avec les ressources disponibles. Ainsi, la figure 4 propose des substitutions réussies, où la sémantique des phrases reste fidèle aux phrases d'origine, grâce aux synonymes comme {*absorption* ; *ingestion*}, {*traitement* ; *prescription*} ou {*traiter* ; *soigner*}. Alors que la figure 5 propose des substitutions non réussies, où la sémantique des phrases n'est pas sauvegardée. Par exemple, la sémantique change dans le cas des synonymes {*corps* ; *mort*}, alors que dans l'exemple avec les synonymes comme {*main* ; *pince*}, {*dents* ; *chicots*} ou {*tête* ; *citron*}, il s'agit de synonymes qui appartiennent à différents niveaux de la langue {*normé* ; *jargon*}. Même si cela ne modifie pas beaucoup la sémantique des phrases, la formulation devient plus familière, ce qui n'etait pas l'effet recherché.

Avant substitution	*Après substitution*
La nourriture n'a pas d'effet sur l'absorption d'anastrozole.	La nourriture n'a pas d'effet sur l'ingestion d'anastrozole.
Vous devez discuter avec votre médecin sur les risques et les options de traitement.	Vous devez discuter avec votre médecin sur les risques et les options de prescription.
Votre médecin peut vous prescrire un médicament visant à prévenir ou traiter cette perte osseuse.	Votre médecin peut vous prescrire un médicament visant à prévenir ou soigner cette perte osseuse.

FIGURE 4 – Exemples de substitutions réussies.

Avant substitution	*Après substitution*
Un abcès est une accumulation de pus sous la peau ou à l'intérieur du corps.	Un abcès est une accumulation de pus sous la peau ou à l'intérieur du mort.
Syndrome du canal carpien (fourmillement, douleur, sensation de froid, faiblesse dans certaines parties de la main).	Syndrome du canal carpien (fourmillement, douleur, sensation de froid, faiblesse dans certaines parties de la pince).

FIGURE 5 – Exemples de substitutions non réussies.

6 Conclusion et perspectives

Dans ce travail, nous avons proposé d'effectuer la simplification automatique de textes médicaux en français. Notre travail propose plusieurs contributions : (1) création de corpus comparables avec des textes médicaux techniques et simplifiés ; (2) alignement manuel de phrases ; (3) observations des procédés de simplification présents dans les corpus ; (4) premiers tests de substitution lexicale ; (5) évaluation des résultats avec trois critères de jugement (grammaticalité, simplification et sémantique).

Nous avons plusieurs perspectives à ce travail : (1) préparer et exploiter un lexique plus approprié pour la substitution lexicale dans les textes médicaux, comme ceux proposés dans les travaux existants (Grabar & Hamon, 2016), ce qui devrait permettre d'augmenter la couverture des substitutions ; (2) mieux gérer l'ambiguïté contextuelle des synonymes, ce qui devrait permettre d'augmenter l'acceptabilité sémantique des substitutions ; (3) augmenter le volume de phrases alignées, ce qui devrait permettre de tester d'autres approches pour la substitution, y compris les approches probabilistes ; (4) combiner différents types de modifications lexicales (substitutions, ajouts de paraphrases et de définitions) ; (5) combiner la simplification lexicale avec la simplification syntaxique pour fournir des résultats plus complets.

Remerciements

La présente publication s'inscrit dans le projet *CLEAR (Communication, Literacy, Education, Accessibility, Readability)* financé par l'ANR sous la référence ANR-17-CE19-0016-01.

Je remercie les relecteurs pour leurs remarques constructives. Je remercie également Natalia Grabar, pour son aide dans la réalisation des travaux décrits ici, ainsi que dans la rédaction de cette publication.

Références

ABUALHAIJA S., MILLER T., ECKLE-KOHLER J., GUREVYCH I. & ZIMMERMANN K.-H. (2017). Metaheuristic approaches to lexical substitution and simplification. In *EACL 2017*, p. 1–11.

AMOIA M. & ROMANELLI M. (2012). SB : mmSystem - using decompositional semantics for lexical simplification. In **SEM 2012*, p. 482–486, Montréal, Canada.

BARBU E., MARTIN-VALDIVIA M., ALFONSO L. & LOPEZ U. (2013). Open book : a tool for helping ASD users' semantic comprehension. In *Proceedings of the 2nd workshop of natural language processing for improving textual accessibility NLP4ITA*, p. 11–19, Atlanta, United States.

BERKMAN N., SHERIDAN S., DONAHUE K., HALPERN D. & CROTTY K. (2011). Low health literacy and health outcomes : An updated systematic review. *Annals of Internal Medicine*, **155**(2), 97–107.

BIRAN O., BRODY S. & ELHADAD N. (2011). Putting it simply : a context-aware approach to lexical simplification. In ACL, Ed., *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies : short papers-Volume 2*, p. 496–501.

BRIN-HENRY F. (2014). Éducation thérapeutique du patient et orthophonie. In *Communiquer malgré l'aphasie*. S. Médical.

BROUWERS L., BERNHARD D., LIGOZAT A.-L. & FRANÇOIS T. (2014). Syntactic Sentence Simplification for French. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR) @ EACL 2014*, p. 47–56, Gothenburg, Sweden.

CHANDRASEKAR R., DORAN C. & SRINIVAS B. (1996). Motivations and methods for text simplification. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, p. 1041–1044, Stroudsburg, PA, USA : Association for Computational Linguistics.

CHMIELIK J. & GRABAR N. (2011). *Détection de la spécialisation scientifique et technique des documents biomédicaux grâce aux informations morphologiques*, In *TAL*, volume 51(2), p. 151–179.

CÔTÉ R. (1996). *Répertoire d'anatomopathologie de la SNOMED internationale, v3.4*. Université de Sherbrooke, Sherbrooke, Québec.

DAVIS T. & WOLF M. (2004). Health literacy : implications for family medicine. *Fam Med*, **36**, 595–598.

DE BELDER J. & MOENS M. (2010). Text simplification for children. In *Workshop on accessible search systems of SIGIR*, p. 1–8.

DEVLIN S. & UNTHANK G. (2006). Helping aphasic people process online information. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '06, p. 225–226, New York, NY, USA : ACM.

DUBAY W. (2004). *The principles of readability*, In *Impact Information*.

FLESCH R. (1948). *A new readability yardstick*, In *Jounal of Applied Psychology*, volume 23, p. 221–233.

FRANÇOIS T., BILLAMI M. B., GALA N. & BERNHARD D. (2016). Automatic ranking of synonyms according to their reading and comprehension difficulty. In *JEP-TALN-RECITAL 2016*, volume 2 of *TALN*, p. 15–28, Paris, France.

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

GASPERIN C., MAZIERO E., SPECIA L., PARDO T. & ALUISIO R. M. (2009). *Natural language processing for social inclusion : a text simplification architecture for different literacy levels*, In *SEMISH-XXXXVI*, p. 397–404.

GLAVAS G. & STAJNER S. (2015). Simplifying lexical simplification : Do we need simplified corpora ? In *ACL-COLING*, p. 63–68.

GOEURIOT L., GRABAR N. & DAILLE B. (2007). *Caractérisation des discours scientifique et vulgarisé en français, japonais et russe*, In *TALN*, p. 93–102.

GRABAR N. & HAMON T. (2016). Exploitation de la morphologie pour l'extraction automatique de paraphrases grand public des termes médicaux. *TAL*, **57**(1), 85–109.

GUNNING R. (1973). *The art of clear writing*. New York, NY : McGraw Hill.

HORN C., MANDUCA C. & KAUCHAK D. (2014). Learning a lexical simplifier using Wikipedia. In *ACL Annual Meeting*, p. 458–463.

INUI K., FUJITA A., TAKAHASHI T., IIDA R. & IWAKURA T. (2003). Text simplification for reading assistance : a project note. In *Proc. of the 2nd international workshop on paraphrasing : paraphrase acquisition and applications*, p. 9–16.

JAUHAR S. & SPECIA L. (2012). UOW-SHEF : SimpLex – lexical simplicity ranking based on contextual and psycholinguistic features. In **SEM 2012*, p. 477–481, Montréal, Canada.

JOHANNSEN A., MARTÍNEZ H., KLERKE S. & SØGAARD A. (2012). Emnlp@cph : Is frequency all there is to simplicity ? In **SEM 2012*, p. 408–412, Montréal, Canada.

JONNALAGADDA S., TARI L., HAKENBERG J., BARAL C. & GONZALEZ G. (2009). Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume : Short Papers*, p. 177–180 : Association for Computational Linguistics.

KOKKINAKIS D. & GRONOSTAJ M. T. (2006). *Comparing lay and professional language in cardiovascual disordes corpora*, In *WSEAS transactions on biology and biomedicine*, p. 429–437.

LIGOZAT A., GROUIN C., GARCIA-FERNANDEZ A. & BERNHARD D. (2012). Annlor : A naïve notation-system for lexical outputs ranking. In **SEM 2012*, p. 487–492.

POPRAT M., MARKÓ K. & HAHN U. (2006). *A language classifier that automatically divides medical documents for experts and health care consumers*, In *MIE 2006 – Proceedings of the XX international congress of the european federation for medical informatics*, p. 503–508. Maastricht.

RELLO L., BAEZA-YATES R. A., BOTT S. & SAGGION H. (2013). Simplify or help ? : text simplification strategies for people with dyslexia. In *W4A*.

SACKETT D. L., ROSENBERG W. M. C., GRAY J. A. M., HAYNES R. B. & RICHARDSON W. S. (1996). Evidence based medicine : what it is and what it isn't. *BMJ*, **312**(7023), 71–72.

SAGGION H. (2017). *Automatic Text Simplification*. Morgan & Claypool Publishers.

SAJOUS F. & HATHOUT N. (2015). GLAWI, a free XML-encoded Machine-Readable Dictionary built from the French Wiktionary. In *Proceedings of the eLex 2015 conference*, p. 405–426, Herstmonceux, England.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees.

SERETAN V. (2012). *Acquisition of Syntactic Simplification Rules for French*. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA). ID : unige :30961.

SHARDLOW M. (2014). Out in the open : Finding and categorising errors in the lexical simplification pipeline. In N. CALZOLARI, K. COUKRI, T. DECLERCK, H. LOFTSSON, B. MAEGAARD, J. MARIANI, A. MORENO, J. ODIJK & S. PIPERIDIS, Eds., *Proc. of the 9th International Conference on Language Resources and Evaluation*, LREC, Reykjavik, Iceland : European Language Resources Association (ELRA).

SINHA R. (2012). Unt-simprank : Systems for lexical simplification ranking. In **SEM 2012*, p. 493–496.

SPECIA L. (2010). *Translating from complex to simplified sentences*, In *International conference on computational processing of the portuguese language (Propor-2010)*, p. 30–39.

SPECIA L., JAUHAR S. & MIHALCEA R. (2012). Semeval-2012 task 1 : English lexical simplification. In **SEM 2012*, p. 347–355.

WILLIAMS S. & REITER E. (2005). Generating readable texts for readers with low basic skills. In *ENLG*.

WOODSEND K. & LAPATA M. (2011). *Learning to simplify sentences with quasi-synchronous grammar and integer programming*, In *EMNLP*, p. 409–420.

WUBBEN S., VAN DEN BOSCH A. & KRAHMER E. (2012). Sentence simplification by monolingual machine translation. In *ACL*, p. 1015–1024.

ZHU Z., BERNHARD D. & GUREVYCH I. (2010). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, p. 1353–1361, Stroudsburg, PA, USA : Association for Computational Linguistics.

Démonstrations

CuriosiText : application web d'aide au peuplement d'ontologies métiers comme ressources lexicales basée sur Word2Vec

Meryl Bothua[1], Delphine Lagarde[1], Laurent Pierre[1]
(1) EDF Lab, 7 Boulevard Gaspard Monge, 91120, Palaiseau
meryl.bothua@edf.fr, delphine.lagarde@edf.fr, laurent.pierre@edf.fr

RESUME

Suite à la mise en place d'une chaîne traitement destinée à extraire automatiquement des actions de maintenance réalisées sur des composants dans des comptes rendus, nous avons cherché à constituer des ressources lexicales à partir de textes souvent mal normalisés sur le plan linguistique. Nous avons ainsi développé une application web, CuriosiText, qui permet de lancer un traitement Word2Vec et de peupler semi automatiquement une ontologie métier avec les termes similaires correctement détectés. Des relations métiers spécifiques peuvent également être ajoutées.

ABSTRACT

CuriosiText: a web application based on Word2Vec helping with the population of ontologies (serving as lexical resources).
After having developed a process dedicated to the extraction of maintenance actions on components from reports, we sought to constitute lexical resources. These reports usually include many lexical irregularities and lack language standardisation. We developed a web application, CuriosiText, based on Word2Vec method that helps to populate an ontology with the similar terms thus detected.

MOTS-CLES : similarité entre mots, plongement de mots, visualisation de données, extraction d'informations, peuplement d'ontologie.

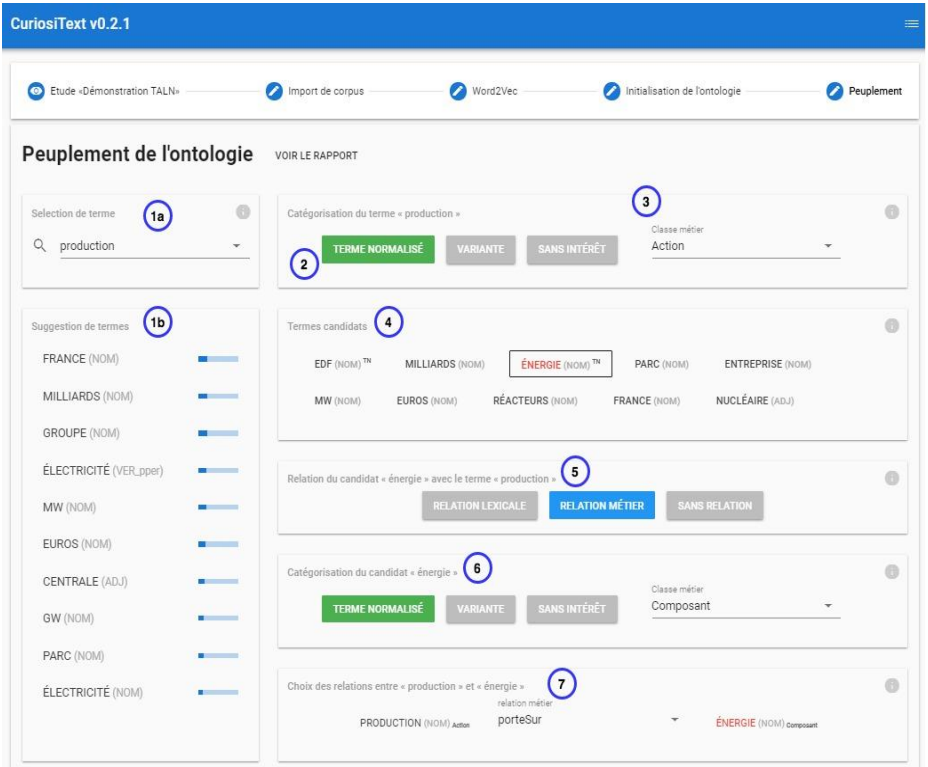
KEYWORDS: word similarity, word embedding, data visualization, information extraction, ontology population.

1 Contexte

Dans le contexte de transition numérique d'EDF et dans sa volonté d'exploiter au mieux ses données, il est aujourd'hui indispensable d'explorer des méthodes pour traiter la masse d'informations textuelles contenues dans les documents techniques, les données relatives à la relation client, ou encore les archives documentaires. Des chaînes de traitements pour de l'extraction de connaissances sont mises en place pour traiter ces contenus textuels. L'automatisation du processus offre un gain de temps conséquent et met en exergue des éléments auparavant noyés dans la masse de données. Afin de rendre pertinent ce type de chaînes de traitement, il est nécessaire de développer en amont différentes ressources lexicales. Afin de faciliter la création de telles ressources, nous avons développé une application web, CuriosiText. Celle-ci intègre la méthode Word2Vec pour extraire des synonymes, des abréviations, des mots mal orthographiés ou encore des phénomènes de multilinguisme. Nous rendons ensuite possible le peuplement d'ontologies que nous utilisons comme ressources lexicales alimentant les chaînes d'extraction de connaissance.

2 L'application CuriosiText

CuriosiText est une application web qui permet de charger des données, de lancer un traitement Word2Vec et de peupler manuellement à partir des termes similaires détectés une ontologie modélisée en amont. Cette ontologie peut être exportée pour être intégrée en tant que ressource lexicale au sein d'une chaîne d'extraction de connaissance. Nous proposerons dans cette démonstration de présenter l'ensemble des fonctionnalités de cet outil pour le peuplement de l'ontologie.



1a L'utilisateur peut requêter un terme (ici « **production** »).

1b Il peut aussi choisir un terme parmi les suggestions (termes les plus fréquents de Word2Vec).

2 Il ajoute une classe lexicale (ici **Terme Normalisé**, le terme étant bien écrit).

3 Il ajoute une classe métier issue de son ontologie métier chargée en amont (ici **Action**).

4 Les candidats de Word2Vec sont retournés et l'utilisateur choisi un terme (ici **« énergie »**).

5 Il spécifie la relation qui lie le terme requêté et le terme candidat (ici **Relation Métier**).

6 Il définit alors la classe lexicale du candidat choisi ainsi que sa classe métier (ici **Terme Normalisé** et **Composant)**. La classe métier est automatiquement héritée du terme requêté si l'on a choisi à l'étape 5 de créer une Relation Lexicale.

7 Il précise enfin la relation métier qui lie les termes (ici **porteSur**). Si l'on a choisi à l'étape 5 de créer une Relation Lexicale, il est possible d'ajouter un lien de synonymie, de variation du terme (cas des abréviations et des fautes d'orthographe) ou de bruit.

Références

MCKEE G. T., MALVERN D. & RICHARDS B. J. (2000). MEASURING VOCABULARY DIVERSITY USING DEDICATED SOFTWARE.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). EFFICIENT ESTIMATION OF WORD REPRESENTATIONS IN VECTOR SPACE. CORR.

MOTIK B., NENOV Y., PIRO R., HORROCKS I. & OLTEANU D. (2014). PARALLEL MATERIALISATION OF DATALOG PROGRAMS IN CENTRALISED, MAIN-MEMORY RDF SYSTEMS. IN C. E. BRODLEY & P. STONE, EDS., PROC. OF THE 28TH AAAI CONF. ON ARTIFICIAL INTELLIGENCE (AAAI 2014), P. 129–137, QUÉBEC CITY, QUÉBEC, CANADA : AAAI PRESS.

SCHMID H. (1994). PROBABILISTIC PART-OF-SPEECH TAGGING USING DECISION TREES. *Insertion/Caractères spéciaux*, onglet *Caractères spéciaux*).

ACCOLÉ : Annotation Collaborative d'erreurs de traduction pour COrpus aLignÉs

Francis Brunet-Manquat[1] Emmanuelle Esperança-Rodier[1]
(1) Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France

Francis.Brunet-Manquat@univ-grenoble-alpes.fr,Emmanuelle.Esperanca-Rodier@univ-grenoble-alpes.fr

RESUME

La plateforme ACCOLÉ (Annotation Collaborative d'erreurs de traduction pour COrpus aLignÉs) propose une palette de services innovants permettant de répondre aux besoins modernes d'analyse d'erreurs de traduction : gestion simplifiée des corpus et des typologies d'erreurs, annotation d'erreurs efficace, collaboration et/ou supervision lors de l'annotation, recherche de modèle d'erreurs dans les annotations.

ABSTRACT

ACCOLÉ: A Collaborative Platform of Error Annotation for Aligned Corpus. We present a platform for the collaborative editing of translation errors. This platform, named ACCOLÉ, offers a range of innovative services that meet the analysis needs of translation errors: simplified management of corpora and typologies of errors, annotation of effective errors, collaboration and/or supervision during annotation, looking for error types in annotations.

MOTS-CLES : Annotations d'erreurs de traduction, Annotation collaborative
KEYWORDS: Annotations of translation errors, Collaborative annotation

La plateforme ACCOLÉ permet l'annotation manuelle des erreurs de traduction selon des critères linguistiques basés sur la typologie de (Vilar et al., 2006). Elle se situe dans la lignée des travaux portant sur l'estimation de la qualité comme QuEst++ (Specia et al., 2015) et de l'analyse d'erreurs tels que Coreference Annotator (Tsoumari et al., 2001) ou BLAST (Stymne, 2011). Les travaux de (Esperança et al., 2016) montrent les limites de ce dernier quant à son utilisation pour la tâche que nous nous sommes fixée. Il s'agit de fournir à un utilisateur une aide dans le choix d'un système de TA à utiliser selon le contexte (compétences linguistiques et informatiques de l'utilisateur, connaissance du domaine du document source à traduire et la tâche pour laquelle il a besoin de traduire le document source.). ACCOLÉ doit donc permettre de détecter quels sont les phénomènes linguistiques qui ne sont pas traités correctement par le système de TA étudié. Nous proposons ainsi, sur la même plateforme une palette de services innovants permettant de répondre aux besoins modernes d'analyse d'erreurs de traduction. Ainsi, les principales fonctionnalités de la plateforme ACCOLÉ sont la gestion simplifiée des corpus, des typologies d'erreurs, des annotateurs, etc. ; l'annotation d'erreurs efficace ; la collaboration et/ou supervision lors de l'annotation ; la recherche de modèle d'erreurs (type d'erreurs dans un premier temps) dans les annotations. La plateforme est disponible en ligne sur un navigateur et ne nécessite aucune installation spécifique.

1 Annotation d'erreurs

La plateforme ACCOLÉ propose de visualiser et d'annoter efficacement les erreurs d'un couple de phrases source/cible. L'annotation se fait en deux étapes. La première étape consiste à sélectionner, à l'aide de la souris, des mots dans la phrase source, et de leur équivalent dans la phrase cible, présentant une erreur de traduction. La seconde étape consiste à choisir le type d'erreur à associer au couple des mots sources/cibles préalablement sélectionnés. En plus de sa simplicité d'usage, ACCOLÉ propose deux mécanismes pour aider l'annotateur dans sa tâche : un mécanisme de supervision permettant à un responsable de contrôler l'avancée de la tâche, ce mécanisme encourage surtout la communication entre superviseur et annotateur par la possibilité de créer des fils de discussion pour un couple de phrases source/cible précis (demander des précisions sur un type d'erreur, pointer une erreur d'annotation, etc.); et un mécanisme collaboratif permettant aux annotateurs de s'entraider ou de discuter autour d'un couple phrase source/cible précis.

2 Représentation des erreurs basée sur les SSTC

La plateforme utilise une représentation des données basée sur les SSTC (Structured String-Tree Correspondences, Boitet et Zaharin 1988). Une erreur est donc constituée d'une étiquette et d'un ensemble de SNODE (intervalle représentant la sous-chaîne dans la phrase source ou cible correspondante). Par exemple dans la figure ci-dessous représentant un exemple d'annotations, l'erreur portant sur "toute l'" et "any" est décrite par son étiquette *Mauvais choix lexical* (cat. *Mot incorrect*, sous-cat. *Sens*), par son positionnement dans la phrase source (SNODE [49-56] - sous chaîne entre le 49ème caractères et le 56ème) et la phrase cible (SNODE [46-48]). L'un des avantages d'utiliser ainsi les SNODE est de se passer d'une structure syntaxique pour décrire l'erreur.

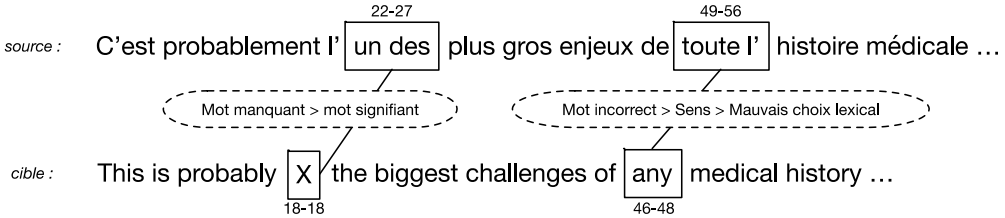


FIGURE 1: exemple d'annotations

3 Recherche d'erreurs

La recherche de type d'erreurs dans les annotations est un atout essentiel de la plateforme. Nous souhaitons l'améliorer en permettant, en plus de l'association d'analyses morphosyntaxiques aux annotations d'erreurs, l'association d'arbres de dépendances, ainsi que l'utilisation d'autres typologies d'erreurs telles que décrites dans Felice (2012) ou bien des métriques multidimensionnelles de qualité (MQM) du projet QT21 (QT21, 2016).

Références

Boitet, C., Zaharin, Y. (1988). Representation trees and string- tree correspondences. *Proc. COLING-88*, 59-64.

Esperança-Rodier E., Didier, J. 2016. Translation Quality Evaluation of MWE from French into English using an SMT system. *Actes de la 38th Conference Translating and the Computer, London, UK, November 17-18, ©2016 AsLing 33–41*

Felice M., Specia L. (2012). Linguistic Features for Quality Estimation. Actes de *7th Workshop on Statistical Machine Translation*, Montréal, Canada, June 7-8, 2012 Association for Computational Linguistics, 96–103

QT21: Quality Translation 21. Available at: http://www.qt21.eu/. Accessed: September 25, 2016.

Specia L., Paetzold G. H. et Scarton C. (2015): Multi-level Translation Quality Prediction with QuEst++. Actes de *ACL-IJCNLP 2015 System Demonstrations*, Beijing, China, 115-120.

Stymne S.(2011). Blast: A Tool for Error Analysis of Machine Translation Output. Actes de *ACL-HLT 2011 System Demonstrations*. June 19-24, 2011. Portland, Oregon, USA, 56-61

Tsoumari, M. and Petasis, G. 2001. Coreference Annotator - A new annotation tool for aligned bilingual corpora. Actes du *Second Workshop on Annotation and Exploitation of Parallel Corpora* (AEPC 2), dans 8th International Conference on Recent Advances in Natural Language Processing (RANLP 2011)

Vilar D., Xu J., D'Haro L. F., Ney H. (2006). Error Analysis of Statistical Machine Translation Output. Actes *LREC*, Genoa, Italy, 697-702

Néonaute, Enrichissement sémantique pour la recherche d'information

Emmanuel Cartier[1], Loïc Galand[1], Peter Stirling[2] et Sara Aubry[2]
(1) Université Paris 13 SPC, LIPN-RCLN, UMR 7030 CNRS, Labex EFL, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse
(2) Bibliothèque nationale de France, Départements du Dépôt légal et des Systèmes d'information, Quai François-Mauriac, 75706 Paris Cedex 13
emmanuel.cartier@lipn.univ-paris13.fr,
loic.galand@lipn.univ-paris13.fr,peter.stirling@bnf.fr,sara.aubry@bnf.fr

Avec l'explosion du nombre de documents numériques accessibles, les besoins en outils pour l'enrichissement sémantique des données textuelles, ainsi que des fonctionnalités avancées de recherche et d'exploration des collections, se font sentir. Cette combinaison entre les domaines de la recherche d'information et du traitement automatique des langues est l'une des caractéristiques du projet Néonaute.

Ce projet, financé par la DGLFLF [1] en 2017 (appel Langues et numérique), regroupe la Bibliothèque nationale de France (BnF), le LIPN - RCLN (CNRS UMR 7030) et l'Université de Strasbourg (LILPA, EA 1339). Son objectif principal est de doter les observateurs de la langue française d'un moteur de recherche s'appuyant sur une collection de sites de presse d'actualité, collectés automatiquement par la BnF au titre de sa mission de dépôt légal de l'internet. Sur cette collection, le projet vise à proposer un moteur de recherche de nouvelle génération, disposant d'une indexation enrichie par l'analyse automatique des textes (analyse morphosyntaxique, entités nommées, thématiques), d'une part, et d'outils de recherche, d'exploration et de visualisation multidimensionnelle interactive des résultats, d'autre part.

Enrichissement des métadonnées par le TAL L'objectif premier du projet est d'enrichir les informations indexées dans le moteur de recherche. La BnF constitue depuis 1996 une archive du web français, qui représente fin 2017 plus de 30 milliards de fichiers et 938 To de données. Depuis décembre 2010, une centaine de sites d'actualités (presse nationale, presse régionale, portails d'information) sont collectés quotidiennement (page d'accueil et liens internes à un clic), constituant la collection dite "Actualités" qui représente 1 milliards de fichiers et 13 To de données. Cette collection est accessible dans les salles de lecture de la BnF via une interface de recherche plein texte (Archives de l'internet Labs) construite à partir du moteur Apache Solr. Le projet Néonaute effectue une analyse *linguistique* automatique des contenus textuels, comme suit :

1. filtrage des contenus collectés pour ne conserver qu'un corpus de pages à contenu textuel : pour ce faire, un certain nombre de filtres ont été développés, aboutissant à ne retenir qu'environ 10% de l'archive totale ;
2. nettoyage des pages filtrées pour ne conserver que le contenu textuel "nouveau" : pour ce faire, un état de l'art et une évaluation nous ont permis de choisir puis d'utiliser la librairie

1. délégation générale à la langue française et aux langues de France

Python *JusText*[2] (Pomikálek, 2011) ;

3. analyse morphosyntaxique du contenu textuel : pour ce faire, après un état de l'art, l'établissement de critères de choix puis une évaluation des précision et rappel sur un échantillon d'une dizaine de pages web, l'outil *Spacy*[3] (Choi *et al.*, 2015) a été retenu ;
4. détection automatique des entités nommées (personnes, lieux, organisations, autres) : la même procédure de sélection a abouti à choisir l'outil *Spacy* ;
5. détection automatique des thématiques de chaque page web : cet aspect est actuellement en cours d'évaluation, avec des techniques d'extraction de mots clés à l'aide d'une pondération des mots avec le modèle TF-IDF, ainsi que des techniques de topic modelling (Allocation de Dirichlet Latente).

Nous présenterons chacune des étapes effectuées, les résultats, leur évaluation et les problèmes non résolus à ce stade du projet.

Exploitation de l'indexation enrichie L'enrichissement *linguistique* des données permet de nouvelles exploitations dans le cadre des moteurs de recherche. En effet, les analyses linguistiques ajoutent minimalement des métainformations liées aux lexies discriminantes, aux entités nommées et aux thèmes principaux de chaque article. Ces enrichissements peuvent être exploités sous forme de *facettes* dans le cadre d'une recherche simple, soit sous forme de *visualisation multidimensionnelle interactive* (Cartier, 2017), pour explorer les données selon différentes points de vue. Par exemple, il sera possible de visualiser l'évolution temporelle des emplois de *twitterisation* sur toute la période, en tenant compte des métadonnées liées à chaque source d'informations (journal, type de presse, auteurs, etc.).

Cas d'usage En plus de l'objectif principal, qui aboutira à la mise à disposition d'un moteur de recherche de nouvelle génération, trois cas d'utilisation sont prévus, dont les premiers résultats seront présentés :

1. étude de l'implantation des néologismes sur une période temporelle de 8 ans (2010-2017) ;
2. étude de l'implantation des préconisations de termes de la DGLFLF ;
3. étude des formes de la féminisation des noms communs dans ce même corpus.

Durant la démonstration, nous présenterons les différentes phases d'analyse linguistique des textes, leur indexation et leur exploitation dans le moteur de recherche, en nous appuyant sur les cas d'utilisation.

Références

CARTIER E. (2017). Neoveille, a Web Platform for Neologism Tracking. In *Proceedings of European Chapter of the Association for Computational Linguistics 2017, Valencia, 3-7 avril 2017*.

CHOI J. D., TETREAULT J. & STENT A. (2015). It depends : Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, volume 1, p. 387–396.

2. https://github.com/miso-belica/jusText
3. https://github.com/explosion/spaCy

POMIKÁLEK J. (2011). *Removing boilerplate and duplicate content from web corpora*. PhD thesis, Masaryk university, Faculty of informatics, Brno, Czech Republic.

Nouveautés de l'analyseur linguistique LIMA

Gaël de Chalendar[1]
(1) CEA,LIST, Laboratoire Vision et Ingénierie des Contenus, Gif-sur-Yvette, F-91191
gael.de-chalendar@cea.fr

RÉSUMÉ

LIMA est un analyseur linguistique libre d'envergure industrielle. Nous présentons ici ses évolutions depuis la dernière publication en 2014.

ABSTRACT

What's New in the LIMA Language Analyzer.

LIMA is a free language analyzer of industrial scope. We present here its evolutions since the last publication in 2014.

MOTS-CLÉS : tokenisation, morphologie, étiquetage morphosyntaxique, analyse syntaxique, entités, relations, interface graphique.

KEYWORDS: tokenization, morphology, PoS tagging, parsing, entities, relations, GUI.

LIMA est l'analyseur linguistique du CEA LIST. Commencé en 2002, il a été présenté à la communauté en 2010 (Besançon *et al.*, 2010) puis placé sous licence libre en 2014 (de Chalendar, 2014) [1].

Depuis la dernière version numérotée 2.1 en 2015, plus de 1200 modifications ont été apportées. La plupart ne sont que des améliorations à la marge, corrections de bugs ou améliorations de l'infrastructure. Nous présentons dans les sections suivantes les changements les plus importants, mais commençons par résumer ci-dessous quelques autres évolutions.

De nouveaux tests unitaires ont été ajoutés. Le système d'intégration continue (IC) a été amélioré avec l'utilisation de conteneurs docker sur les plateformes Semaphore, Appveyor et Travis. Nous utilisation désormais le système des release github pour distribuer les paquets générés par l'IC. Nous avons aussi amélioré la construction multiplateforme en utilisant le système de construction Ninja sur l'ensemble d'entre elles. Concernant les aspects TAL, nous avons débuté la transition vers l'utilisation d'étiquettes issues du projet Universal Dependencies. Enfin, nous utilisons désormais SVMTool comme étiqueteur morphosyntaxique par défaut.

Support du portugais

Nous avons ajouté à LIMA le support de la langue portugaise en utilisant le dictionnairs Delaf PB (sous licence LGPL) et le corpus annoté Mac-Morpho (sous licence CC BY 4.0). Nos derniers résultats d'évaluation de l'étiquetage morphosyntaxique par validation croisée sur le corpus d'apprentissage donnent une précision de 96%, du niveau de l'état de l'art. Il nous reste désormais à ajouter une prise en compte des expressions idiomatiques, le traitement des entités nommées et des règles d'analyse syntaxique pour avoir les mêmes fonctionnalités que dans les autres langues.

1. https://github.com/aymara/lima

Entités nommées améliorées

Nous avons amélioré nos traitements des entités nommées selon deux axes. D'une part, nous avons ajouté un module permettant d'effectuer une recherche approximative, ce qui permet de repérer des noms malgré des formes variables. Quand un dictionnaire de référence existe, on admet des erreurs (suppression ou ajout de caractères) et on utilise des motifs de généralisation. On peut alors trouver les noms du dictionnaire avec une marge d'erreur spécifiée. D'autre part, nous avons ajouté un module de reconnaissance statistique des entités à base de CRF, fondé sur la bibliothèque Wapiti. Celui-ci nous a permis d'obtenir d'excellents résultats, sur certains types d'entités, dès lors qu'un corpus annoté est disponible.

Interface graphique

Notre ambition est de rendre LIMA accessible à tous, aussi bien des industiels désirant intégrer des traitements de TAL dans leurs applications que des étudiants en linguistique devant aborder le TAL. Pour ces derniers et tout autre utilisateur occasionnel, nous avons développé une interface graphique permettant d'accéder aux principales fonctionnalités de LIMA. Celle-ci ne permet pour le moment que de lancer une analyse et de consulter les résultats sous divers formats. Elle sera enrichie à l'avenir avec de nouveaux outils de visualsation et une interface de configuration.

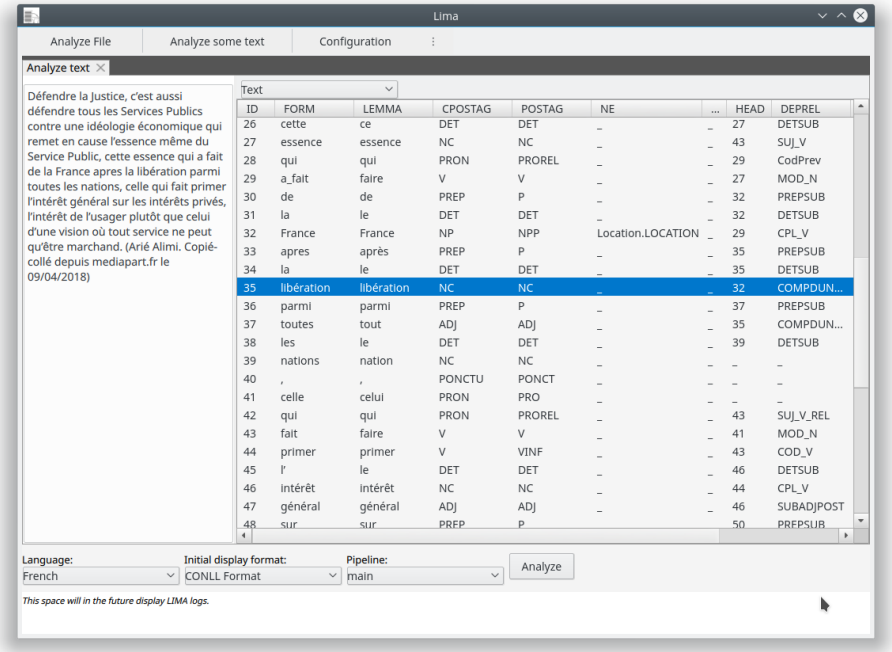


FIGURE 1 – L'interface graphique de LIMA

Conclusion

LIMA ne reste pas en marge de la vague de fond qui révolutionne le domaine du TAL, les approches neuronales. Déjà, un module neuronal d'extraction d'entités nommées a été ajouté et un module d'analyse syntaxique est en cours de finalisation de son intégration. Au delà de ces deux modules, le travail sur l'infrastructure nécessaire pour générer de bout en bout des analyseurs à partir de corpus annotés est en cours. Cela nous permettra de participer aux prochaines occurrences de la tâche partagée CoNLL. Pour autant, nous voulons conserver les fonctionnalités qui font la spécificité de LIMA : multiplateforme, performance, facilité d'usage et intégrabilité dans des outils industiels.

Références

BESANÇON R., CHALENDAR (DE) G., FERRET O., GARA F. & SEMMAR N. (2010). LIMA : A Multilingual Framework for Linguistic Analysis and Linguistic Resources Development and Evaluation. In *Proceedings of Language Resources and Evaluation Conference, 2010*, Malta.

DE CHALENDAR G. (2014). The LIMA Multilingual Analyzer Made Free : FLOSS Resources Adaptation and Correction. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, p. 2932–2937.

Un outil d'étiquetage rapide et un corpus libre en entités nommées du Français

Yoann Dupont[1]
(1) Laboratoire Lattice (CNRS, ENS, Université Sorbonne Nouvelle, PSL Research University, USPC)
1 rue Maurice Arnoux, 92120 Montrouge
yoa.dupont@gmail.com

RÉSUMÉ

Dans cet article, nous présentons un outil pour effectuer l'étiquetage rapide de textes bruts. Il peut charger des documents annotés depuis divers formats, notamment BRAT et GATE. Il se base sur des raccourcis claviers intuitifs et la diffusion d'annotation à l'échelle du document. Il permet d'entraîner des systèmes par apprentissage que l'on peut alors utiliser pour préannoter les textes.

ABSTRACT

A fast tagging tool and a free French named entity corpus

In this article we present a tool for fast tagging of raw texts. It handles multiple input and output formats, such as BRAT and GATE. For fast tagging, the tool relies on intuitive keyboard shortcut and document-wide annotation broadcasting. The tools allows to train machine learning systems that can be used to preannotate texts.

MOTS-CLÉS : étiquetage, entités nommées, corpus, annotation structurée, GUI.

KEYWORDS: tagging, named entities, corpus, structured tagging, GUI.

1 Introduction

Pour de nombreuses tâches de TAL, des textes annotées sont capitaux mais demeurent trop peu nombreux, ou ont une licence restrictive. Il existe déjà de nombreux outils pour annoter des textes bruts, parmi lesquels nous pouvons citer GATE (Cunningham *et al.*, 2013) ou BRAT (Stenetorp *et al.*, 2012). Ces outils ont cependant deux inconvénients principaux : le premier est d'être plutôt lents pour annoter et le second est qu'ils ne gèrent qu'un format, le leur. Pour cette raison, nous proposons ici un outil d'annotation rapide et capable de gérer des données de formats divers. L'outil que nous présentons est un module de SEM (Dupont, 2017).

2 L'outil et le corpus "preuve de concept"

L'outil que nous présentons a été conçu pour annoter rapidement dans le cadre de tâches comme l'étiquetage morphosyntaxique ou la reconnaissance d'entités nommées, mais peut se montrer utile pour toute tâche où des empans textuels doivent être annotés. Il est écrit en python en utilisant la librairie Tkinter (Shipman, 2013) Il permet d'annoter un corpus document par document. Afin

d'améliorer la vitesse d'annotation l'outil recourt à des raccourcis claviers déduits du jeu d'annotation qui doit être chargé (plusieurs jeux peuvent être gérés de manière indépendante). Si le jeu d'annotation contient un type "lieu", son raccourci par défaut sera "l". Nombre d'éléments peuvent se trouver répétés à de nombreuses reprises dans le texte. Par exemple, annoter toutes les occurrences d'une même personnes d'un roman peut s'avérer fastidieux et sujet à l'erreur s'il faut annoter les éléments un à un. Pour combler ce problème, si l'utilisateur souhaite annoter un élément textuel, il peut diffuser l'annotation à l'échelle du document. Cette opération n'est pas sans source d'erreur, il n'est pas impossible que certains "Rennes" annotés soient en fait "Inria de Rennes". Pour gérer ce cas, l'outil propose d'explorer l'historique des annotations, classées par date décroissante, effectuées par l'utilisateur afin de les réviser. Une autre source de vitesse est dans l'utilisation de données préannotatées et dans l'apprentissage de systèmes à partir des données annotées. L'outil propose actuellement d'entraîner des CRF (Lafferty *et al.*, 2001) à l'aide de Wapiti (Lavergne *et al.*, 2010) et prévoit d'intégrer l'entraînement de modèles neuronaux. À terme, il proposera d'entraîner des systèmes sur d'autres tâches, où les annotations ont une structure arborée.

Nous proposons, en preuve de concept, un corpus annoté en entités nommées. Le jeu d'annotation comprend les types suivants : les lieux ("Rennes", "la lune", "la Loire", etc.) ; les personnes ("Emmanuel Macron") ; les organisations (sans distinction entre les organisations et entreprises) ; les dates ("lundi 14 mai 2018", "mai 2018", mais pas "hier") ; les heures absolues ("midi", "14 heures", mais pas "le soir") ; les objets ("satellite James Webb", "la station spatiale internationale", etc.) ; les évènements ("CORIA-TALN", mais aussi les évènements climatiques comme "la tempête Egon"). Nous avons constitué le corpus en récupérant les articles de de la partie française de Wikinews pour l'année 2017. Le contenu de chaque document ne comprend que le titre et le corps de chaque article, nous avons ignoré les images avec légende ainsi que les sources citées. Nous avons également supprimé les documents qui énumèrent des résultats sportifs. Le corpus comprend à l'heure actuelle environ 7300 annotations structurées (par exemple "Tour de France" est un évènement, mais "France" est également annoté en tant que lieu), ce qui représente environ 7 heures de travail, pour une cadence moyenne d'environ 1000 annotations par heure (hors phases d'apprentissage et d'annotation automatique).

Références

CUNNINGHAM H., TABLAN V., ROBERTS A. & BONTCHEVA K. (2013). Getting more out of biomedical documents with gate's full lifecycle open source text analytics. *PLoS computational biology*, **9**(2), e1002854.

DUPONT Y. (2017). Exploration de traits pour la reconnaissance d'entités nommées du Français par apprentissage automatique. In *Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL)*, p. 42–55.

LAFFERTY J., MCCALLUM A. & PEREIRA F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*, p. 282–289.

LAVERGNE T., CAPPÉ O. & YVON F. (2010). Practical very large scale CRFs. In *Proceedings of ACL'2010*, p. 504–513 : Association for Computational Linguistics.

SHIPMAN J. W. (2013). Tkinter 8.4 reference : a gui for python.

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

PyRATA, Python Rule-based feAture sTructure Analysis

Nicolas Hernandez
LS2N, Université de Nantes, France
nicolas.hernandez@univ-nantes.fr

MOTS-CLÉS : analyse à base de règles, annotation sémantique, expression régulière, extraction d'information, fouille de texte, Python 3.

KEYWORDS: rules-based analysis, semantic annotation, regular expression, information extraction, text mining, Python 3.

1 Résumé

Les approches à base de règles ne doivent pas être opposées à celles à base d'apprentissage automatique. Les premières sont connues pour permettre d'obtenir rapidement des résultats auto-explicatifs avec seulement quelques règles, mais présentent l'inconvénient d'être difficiles à maintenir lorsque les règles deviennent complexes et que leur nombre croît. Les approches à base d'apprentissage sont capables de généralisation, et donc, de fournir des résultats même sur des données jamais rencontrées, mais elles requièrent des données d'entraînement lesquelles résultent souvent d'une tâche d'annotation manuelle coûteuse, et leurs résultats sont plus difficiles à expliquer. Actuellement les approches à base de règles ne sont pas populaires dans la communauté du Traitement Automatique des Langues (TAL). Néanmoins il y a encore de bonnes raisons de les utiliser : 1) Puisqu'elles ne requièrent pas de données d'entraînement, elles constituent une bonne première approche pour explorer des données et définir plus précisément un problème ; 2) Pour certaines langues, des problèmes peuvent être traîtées de manière déterministe ; 3) Pour produire des données qui serviront à entraîner un système à base d'apprentissage ; 4) Pour extraire des traits des données et laisser un système à base d'apprentissage apprendre à les combiner ; 5) Pour augmenter les capacités d'un système à base d'apprentissage en pré- ou post-traitant les données afin d'obtenir une performance de 100 % (Manning, 2011).

La communauté du TAL bénéficie de quelques solutions logicielles [1] pour rechercher des motifs d'annotations et traiter les résultats, à savoir : *GATE JAPE* [2] (Cunningham *et al.*, 1999) et *UIMA RUTA* [3] (Kluegl *et al.*, 2016). Ces solutions requièrent une prise en main de leur langage d'expression des règles, et s'insèrent dans l'adoption d'un cadre d'analyse textuelle plus global, qui a son tour requière la prise en main de ses concepts ainsi que quelques bagages techniques (RUTA est très intégré à l'environnement de développement Eclipse). De plus, le programmeur aura préférentiellement à développer en Java pour utiliser ces solutions. Les utilisateurs du langage de programmation Python ne bénéficient pas d'instruments aussi avancés, mêmes si quelques modules sont disponibles à savoir :

1. Nous ne considérons pas ici les environnements tels que l'*IMS Open Corpus Workbench (CWB) Corpus Query Processing (CQP)* (Evert & Hardie, 2011), Nooj (Silberztein, 2005) `http://www.nooj4nlp.net` ou Unitex (Paumier *et al.*, 2009) `http://unitexgramlab.org` qui sont très ancrés dans une analyse linguistique.
2. `https://gate.ac.uk/sale/tao/splitch8.html`, Java 8, GNU
3. `https://uima.apache.org/ruta.html`, Java 8, Apache v2

Python nltk chunk (Bird, 2006), *clips pattern.search* (De Smedt & Daelemans, 2012) et *spaCy*. Le module spaCy présente les meilleures performances en temps de traitement mais a une expressivité très limitée et requière des compétences en programmation. Le module pattern se focalise sur certains types d'annotation et contraint à l'usage de ses traitements linguistiques. Le module pattern n'autorise pas d'exprimer des motifs sur plus d'un type d'annotation à la fois.

Nous présentons *PyRATA* (*P*ython *R*ules-based fe*A*ture s*T*ructure *A*nalysis) un module Python (version 3) diffusé sous licence Apache V2 et disponible sur github [4] et dans les dépots `pypi` [5]. PyRATA a pour objectif de permettre de l'analyse à base de règles sur des données structurées. Le langage de PyRATA offre une expressivité qui couvre les fonctionnalités proposées par les modules alternatifs et davantage. Conçu pour être intuitif, la syntaxe des motifs et l'interface de programmation (API) suivent les définitions de standards existants, respectivement la syntaxe des expressions régulières de Perl et l'API du module Python `re`. PyRATA travaille sur des structures de données simples et natives de Python : une liste de dictionnaires (c-à-d une liste de tables d'associations). Cela lui permet de traiter des données de différentes natures (textuelles ou non) telles qu'une liste de mots, une liste de phrases, une liste de messages d'un fil de discussion, une liste d'événements d'un agenda... Cette spécificité le rend indépendant de la nature des annotations (a fortiori linguistiques) associées à la donnée manipulée. Ce travail a été financé par le projet ANR 2016 PASTEL [6].

Références

BIRD S. (2006). Nltk : The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL '06, p. 69–72, Stroudsburg, PA, USA : Association for Computational Linguistics.

CUNNINGHAM H., CUNNINGHAM H. & TABLAN V. (1999). Jape : a java annotation patterns engine.

DE SMEDT T. & DAELEMANS W. (2012). Pattern for python. *Journal of Machine Learning Research*, **13**, 2063–2067.

EVERT S. & HARDIE A. (2011). Twenty-first century corpus workbench : Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 conference*, University of Birmingham, UK.

KLUEGL P., TOEPFER M., BECK P.-D., FETTE G. & PUPPE F. (2016). Uima ruta : Rapid development of rule-based information extraction applications. *Natural Language Engineering*, **22**, 1–40.

MANNING C. D. (2011). *Part-of-Speech Tagging from 97% to 100% : Is It Time for Some Linguistics ?*, In A. F. GELBUKH, Ed., *Computational Linguistics and Intelligent Text Processing : 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part I*, p. 171–189. Springer Berlin Heidelberg : Berlin, Heidelberg.

PAUMIER S., NAKAMURA T. & VOYATZI S. (2009). UNITEX, a Corpus Processing System with Multi-Lingual Linguistic Resources. In *eLexicography in the 21st century : new challenges, new applications (eLEX'09)*, p. 173–175.

SILBERZTEIN M. (2005). Nooj : A linguistic annotation system for corpus processing. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, HLT-Demo '05, p. 10–11, Stroudsburg, PA, USA : Association for Computational Linguistics.

4. `https://github.com/nicolashernandez/PyRATA`
5. `https://pypi.python.org/pypi/PyRATA`
6. `http://www.agence-nationale-recherche.fr/?Projet=ANR-16-CE33-0007`

Un corpus en arabe annoté manuellement avec des sens WordNet

Marwa Hadj Salah[1, 2] Hervé Blanchon[1] Mounir Zrigui[2] Didier Schwab[1]
(1) Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP*, LIG, 38000 Grenoble, France
(2) LaTICE, Tunis, 1008, Tunisie
Prénom.Nom@univ-grenoble-alpes.fr, Prénom.Nom@fsm.rnu.tn

RÉSUMÉ

OntoNotes comprend le seul corpus manuellement annoté en sens librement disponible pour l'arabe. Elle reste peu connue et utilisée certainement parce que le projet s'est achevé sans lier cet inventaire au *Princeton WordNet* qui lui aurait ouvert l'accès à son riche écosystème. Dans cet article, nous présentons une version étendue de OntoNotes Release 5.0 que nous avons créée en suivant une méthodologie de construction semi-automatique. Il s'agit d'une mise à jour de la partie arabe annotée en sens du corpus en ajoutant l'alignement vers le *Princeton WordNet* 3.0. Cette ressource qui comprend plus de 12 500 mots annotés est librement disponible pour la communauté. Nous espérons qu'elle deviendra un standard pour l'évaluation de la désambiguïsation lexicale de l'arabe.

ABSTRACT

Arabic Manually Sense Annotated Corpus with WordNet Senses

OntoNotes is the only Arabic Manually Annotated Corpus freely available for the Arabic language. It remains little known and exploited certainly because the project ended without linking this inventory to *Princeton WordNet* which would have given it access to its rich ecosystem. In this article, we present an extended version of OntoNotes Release 5.0 that we created using a semi-automatic construction methodology. This is an update of the Arabic part of the sense-annotated corpus by adding the alignment to the *Princeton WordNet* 3.0. This resource that includes more than 12,500 annotated words will be freely available for the community. We hope that it will become a standard for the evaluation of the lexical disambiguation of Arabic.

MOTS-CLÉS : Corpus annoté en sens, langue arabe, alignement de sens interlingues.

KEYWORDS: Sense annotated corpus, arabic language, interlingual sense alignment.

1 OntoNotes Release 5.0

Le projet OntoNotes (Weischedel *et al.*, 2013) est le résultat d'un travail collaboratif entre *BBN Technologies*, l'Université du Colorado, l'Université de Pennsylvanie et l'Institut des sciences de l'information de l'Université de Californie du Sud. *OntoNotes Release 5.0* est la dernière version proposée par ce projet. C'est un grand corpus annoté libre de droit, construit à 90% d'accord inter-annotateur avec des informations structurelles (syntaxe et structures prédicat-arguments) et sémantiques superficielles (sens du mot lié à une ontologie et co-référence). Le corpus contient plusieurs genres de textes en anglais et chinois et uniquement des données News pour la partie arabe.

*. Institute of Engineering Univ. Grenoble Alpes

2 Enrichissement de la partie arabe de l'OntoNotes Release 5.0

Les parties anglaises et chinoises de *OntoNotes Release 5.0* sont annotées avec des sens issus du *Princeton WordNet*. Malheureusement, le projet n'a pas pu être mené jusqu'au bout sur la partie arabe et le lien entre les annotations *OntoNotes* et le *Princeton WordNet* sont absentes. Nous proposons ici une mise à jour de la partie arabe de *OntoNotes Release 5.0* d'une manière semi-automatique pour obtenir des mots annotés en sens avec le *Princeton WordNet* 3.0. La figure 1 présente l'architecture globale de la partie arabe de l'Ontonote.

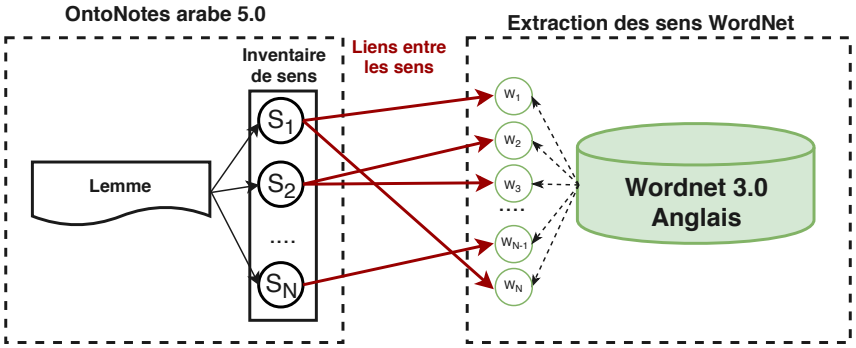


FIGURE 1 – Architecture globale de l'*OntoNotes Release 5.0* après l'ajout des correspondances pour les 261 lemmes uniques de leurs sens vers ceux du *Princeton WordNet* 3.0

Ce traitement semi-automatique d'annotations et de vérification réalisé s'est avéré coûteux en temps (quatre mois.hommes de travail). Le tableau 1 présente la description d'*OntoNotes Release 5.0* ainsi que le nombre de correspondances$_{WordNet}$ uniques ajoutées.

	#Lemmes	#Lemmes uniques	#Sens uniques	**#Correspondances$_{WordNet}$ uniques**
Verbes	3990	150	642	**4182**
Noms	8534	111	463	**1376**
Total	12524	261	1105	**5558**

TABLE 1 – Description d'*OntoNotes Release 5.0* après l'ajout des correspondances vers le *Princeton WordNet* 3.0

La version étendue de l'*OntoNote 5.0* sera disponible pour la communauté et pourra être utilisée dans plusieurs applications du traitement automatique du langage naturel pour la langue arabe, notamment dans la tâche de désambiguïsation lexicale. Avant ce travail, il n'existait aucun corpus en arabe manuellement annoté en sens pour la langue arabe qui soit librement disponible. Cette ressource facilitera la comparaison et/ou la construction de systèmes de désambiguïsation lexicale pour cette langue.

Références

F. BENARMARA, N. HATOUT, P. MULLER & S. OZDOWSKA, Eds. (2007). *Actes de TALN 2007 (Traitement automatique des langues naturelles)*, Toulouse. ATALA, IRIT.

G. DIAS, Ed. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

LAIGNELET M. & RIOULT F. (2009). Repérer automatiquement les segments obsolescents à l'aide d'indices sémantiques et discursifs. In *Actes de TALN 2009 (Traitement automatique des langues naturelles)*, Senlis : ATALA LIPN.

LANGLAIS P. & PATRY A. (2007). Enrichissement d'un lexique bilingue par analogie. In (Benarmara *et al.*, 2007), p. 101–110.

SERETAN V. & WEHRLI E. (2007). Collocation translation based on sentence alignment and parsing. In (Benarmara *et al.*, 2007), p. 401–410.

WEISCHEDEL R., PALMER M., MARCUS M., HOVY E., PRADHAN S., RAMSHAW L., XUE N., TAYLOR A., KAUFMAN J., FRANCHINI M., EL-BACHOUTI M., BELVIN R. & HOUSTON A. (2013). Ontonotes release 5.0. *LDC2013T19. Web Download. Philadelphia : Linguistic Data Consortium*.

Articles DeFT

DEFT2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en Île de France

Patrick Paroubek[1] Cyril Grouin[1] Patrice Bellot[2] Vincent Claveau[3]
Iris Eshkol-Taravella[4] Amel Fraisse[5] Agata Jackiewicz[6] Jihen Karoui[7]
Laura Monceaux[8] Juan-Manuel Torres-Moreno[9]
(1) LIMSI, CNRS, Université Paris-Saclay, F-91405 Orsay
(2) LSIS, Aix Marseille Université, CNRS, F-13397 Marseille
(3) CNRS, IRISA, Université de Rennes, F-35042 Rennes
(4) MoDyCo, Université Paris Nanterre, CNRS, F-92001 Nanterre
(5) GERIICO, Université de Lille, F-59653 Villeneuve-d'Ascq
(6) Praxiling, Université Paul Valéry Montpellier 3, CNRS, F-34199 Montpellier
(7) LIUM, Le Mans Université, F-72085 Le Mans
(8) LS2N, Université de Nantes, CNRS, Ecole centrale de Nantes, IMT Atlantique, F-44322 Nantes
(9) LIA, Université d'Avignon et des Pays de Vaucluse, F-84911 Avignon
{pap,grouin}@limsi.fr, patrice.bellot@lis-lab.fr,
vincent.claveau@irisa.fr, ieshkolt@parisnanterre.fr,
amel.fraisse@univ-lille3.fr, agata.jackiewicz@univ-montp3.fr,
jihen.karoui@univ-lemans.fr, Laura.Monceaux@univ-nantes.fr,
juan-manuel.torres@univ-avignon.fr

RÉSUMÉ

Cet article présente l'édition 2018 de la campagne d'évaluation DEFT (Défi Fouille de Textes). A partir d'un corpus de tweets, quatre tâches ont été proposées : identifier les tweets sur la thématique des transports, puis parmi ces derniers, identifier la polarité (négatif, neutre, positif, mixte), identifier les marqueurs de sentiment et la cible, et enfin, annoter complètement chaque tweet en source et cible des sentiments exprimés. Douze équipes ont participé, majoritairement sur les deux premières tâches. Sur l'identification de la thématique des transports, la micro F-mesure varie de 0,827 à 0,908. Sur l'identification de la polarité globale, la micro F-mesure varie de 0,381 à 0,823.

ABSTRACT

DEFT2018 : Information Retrieval and Sentiment Analysis in Tweets about Public Transportation in Île de France Region

This paper presents the 2018 DEFT text mining challenge. From a corpus of tweets, four tasks were proposed : first, to identify tweets about public transportation ; second, based on those tweets, to identify the global polarity (negative, neutral, positive, mixed), to identify clues of sentiment and target, and to annotate each tweet in terms of source and target concerning all expressed sentiments. Twelve teams participated, mainly on the two first tasks. On the identification of tweets about public transportation, micro F-measure values range from 0.827 to 0.908. On the identification of the global polarity, micro F-measure values range from 0.381 to 0.823.

MOTS-CLÉS : Classification automatique, Analyse de sentiments, Fouille de texte.

KEYWORDS: Automatic Classification, Sentiment Analysis, Text Mining.

1 Introduction

Dans la continuité de l'édition 2015 (Hamon *et al.*, 2015), la treizième édition du DÉfi Fouille de Textes (DEFT) porte sur l'extraction d'information et l'analyse de sentiments dans des tweets rédigés en français sur la thématique des transports. La campagne s'est déroulée sur une période limitée avec une ouverture des inscriptions le 31 janvier, la diffusion des données d'entraînement à partir du 19 février, et le déroulement de la phase de test entre le 28 mars et le 5 avril, sur une durée de trois jours fixée par chacun des participants. Quinze équipes se sont inscrites, dont une hors de France (Canada) et quatre issues d'entreprises privées. Au final, douze équipes auront participé (voir tableau 1).

Équipe	Nom de l'équipe	Affiliation	T1	T2	T3	T4
E1	EDF R&D	EDF R&D (entreprise)	X	X		
E2	CLaC	CLaC, Université Concordia	X	X		
E3	Tweetaneuse	STIH, Sorbonne Université ; LIPN, Université Paris 13	X	X		
E5	Synapse IRIT	Synapse Développement (entreprise) ; IRIT, Université Toulouse 3 Paul Sabatier	X	X		
E6	IRISA	IRISA, INSA Rennes	X	X	X	
E7	LIP6	LIP6, Sorbonne Université	X	X		
E8	Eloquant	Eloquant (entreprise)	X	X		
E9	EPITA	EPITA	X	X		
E10	UTTLM2S	Université Technologique de Troyes	X	X		
E11	Syllabs	Syllabs (entreprise)	X			
E14	ADVTeam	LIRMM, Université Montpellier, CNRS	X	X		
E15	LIS Lab	LIS, Aix Marseille Université		X		

TABLE 1 – Participation des équipes inscrites aux différentes tâches de la campagne DEFT2018

2 Corpus

Le corpus est constitué de tweets en français qui portent sur les transports en Île-de-France. Il contient 76 732 tweets sélectionnés parmi 80 000 tweets annotés manuellement. Les messages sont issus d'une sélection à base de mots-clés et de mesure d'entropie pour filtrer les doublons et les messages dépourvus de texte intelligible.

Chaque message a été annoté parmi cinq types de groupes et quatre types de relations.

— Groupes : la *SOURCE* est la séquence de mots faisant référence à l'entité qui exprime une subjectivité, la *CIBLE* est la séquence de mots faisant référence à l'entité sur laquelle porte cette subjectivité, l'*EXPRESSION d'OPINION/SENTIMENT/EMOTION (OSEE)* prend une valeur parmi les 18 catégories sémantiques polarisées proposés dans Fraisse & Paroubek (2014), le *MODIFIEUR* est marqueur d'intensité de l'expression de subjectivité et la *NEGATION* est un marqueur de négation

— Relations : *DIT* entre la source et l'OSEE, *SUR* entre l'OSEE et la cible, y compris les cibles intermédiaires, *MOD* entre le modifieur et l'OSEE, et *NEG* entre la négation et l'OSEE ou les cibles intermédiaires.

3 Description des tâches

3.1 Présentation

Autour de la thématique des transports, et sur la base des annotations précédentes, nous proposons quatre tâches, dont une très exploratoire sur l'analyse de sentiment à granularité sémantique fine.

Tâche 1 : classification transport/non-transport La première tâche vise à déterminer si un message concerne les transports ou non. Même si le message ne fait référence aux transports que de manière secondaire ou contextuelle, il sera considéré comme relatif aux transports.

— Transport : *Les gars qui puent des aisselles dans le bus c'est vous*
— Autre : *@InfoAbonneCanal bjr j' ai 2 décodeur canal , un me demande d' insérer la carte alors que dans le second les 2 carte fonctionne*

Tâche 2 : polarité globale La deuxième tâche consiste à déterminer la polarité globale d'un message concernant obligatoirement les transports, parmi quatre classes :

— POSITIF (message positif) : *J' ai trouvé une carte navigo dans le bus j' espère que la dame qui l' avait à Facebook sinon je vais pas pouvoir lui envoyer par la poste*
— NEGATIF (message négatif) : *Les gars qui puent des aisselles dans le bus c'est vous*
— NEUTRE (message factuel et objectif) : *Y' a une meuf elle a prit le bus pour s' arrêter à l' arrêt d' après , ils sont à 2 minutes l' un de l' autre à pied*
— MIXPOSNEG (message contenant des expressions positives et négatives, mais aucune des deux polarités ne domine) : *Bon voyage en mégabus malgré le retard le dimanche matin lyon marseille très bon bus manque un peu de confort le retour nikel le 7février*

Tâche 3 : marqueur de sentiment et cible Pour un message sur les transports exprimant des sentiments, cette tâche vise à déterminer, pour chaque expression du message : (i) l'empan de texte minimal qui renvoie à l'expression de sentiment, à l'exclusion des modifieurs et adjoints, et (ii) l'empan de texte maximal qui renvoie à la cible du sentiment, c'est-à-dire l'objet qu'il concerne, y compris modifieurs et adjoints.

Tâche 4 (exploratoire) : annotation complète Étant donné un message concernant les transports et exprimant des sentiments, la quatrième tâche vise à déterminer pour chaque expression de sentiment l'empan de texte minimal référant à l'expression de sentiment et les empans de texte maximaux référant respectivement à la CIBLE du sentiment (l'objet qu'il concerne) et à la SOURCE (l'entité qui exprime ce sentiment). Le cas échéant, on indiquera aussi les empans de texte minimaux en relation avec l'expression de sentiment qui référent à une cible ou un dérangement (voir figure 1).

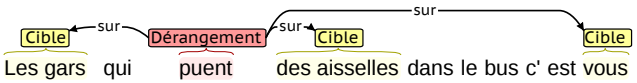


FIGURE 1 – Relations « sur » de l'expression de sentiment (catégorie « dérangement ») vers la cible

3.2 Statistiques

Nous présentons dans le tableau 2 de la distribution des 76 732 tweets annotés dans les corpus d'entrainement et de test, en fonction de la répartition entre catégories transport et non-transport utilisées dans la tâche 1 et de la polarité globale des tweets (POSITIF, NEGATIF, NEUTRE, MIXPOSNEG) utilisés dans la tâche 2 (seuls les tweets de la thématique des transports sont annotés en polarités).

	Transport		Autre		Total
	Entrainement	Test	Entrainement	Test	
POSITIF	7 328	857	*Pas d'annotation des polarités si autre thématique*		8 185
NEGATIF	13 109	1 525			14 634
NEUTRE	12 611	1 304			13 915
MIXPOSNEG	2 420	255			2 675
TOTAL	35 468	3 941	33 448	3 875	76 732

TABLE 2 – Distribution des tweets par corpus (entrainement, test), en fonction de la répartition entre catégories transport et non-transport, et de la polarité (POSITIF, NEGATIF, NEUTRE, MIXPOSNEG). Seuls les tweets de la thématique des transports sont annotés en polarités

4 Résultats

4.1 Mesures d'évaluation

Dans le cadre des deux premières tâches, nous utilisons les mesures habituelles de rappel, précision et F-mesure, calculées au moyen des micro-mesures (Manning & Schütze, 2000) : micro-rappel (formule 1, avec n le nombre total de classes) et micro-précision (formule 2, avec n le nombre total de classes), ainsi que la micro F-mesure calculée sur la base des résultats des deux précédentes mesures. Les micro-mesures attribuent un poids équivalent à chaque élément mesuré, indépendamment de la classe d'appartenance de cet élément. Les systèmes privilégiant les classes à fort effectif sont donc privilégiés dans ce mode de calcul, par opposition aux systèmes essayant de couvrir chacune des classes et notamment les classes à faible effectif.

$$\text{Micro-rappel} = \frac{\sum_{i=1}^{n} \text{vrais positifs}(i)}{\sum_{i=1}^{n} \text{vrais positifs}(i) + \sum_{i=1}^{n} \text{faux négatifs}(i)} \quad (1)$$

$$\text{Micro-précision} = \frac{\sum_{i=1}^{n} \text{vrais positifs}(i)}{\sum_{i=1}^{n} \text{vrais positifs}(i) + \sum_{i=1}^{n} \text{faux positifs}(i)} \quad (2)$$

4.2 Tâche 1 : classification transport/non-transport

Nous présentons dans le tableau 3 les décomptes en termes de vrais positifs, faux positifs et faux négatifs, ainsi que les valeurs de précision, rappel et F-mesure calculées au moyen des micro-mesures sur chacune des soumissions des participants à la tâche 1, classées par F-mesure décroissante. Nous indiquons le rang global de chaque équipe sur la base de la meilleure soumission. La figure 2 fournit

Rang global	Équipe et soumission	Décomptes			Micro-mesures		
		VP	FP	FN	Précision	Rappel	F-mesure
1	E3.R2 – Tweetaneuse	6497	1319	0	0,83124	1,00000	0,90785
2	E7.R4 – LIP6	6491	1325	0	0,83048	1,00000	0,90739
–	E7.R2 – LIP6	6490	1326	0	0,83035	1,00000	0,90731
–	E7.R5 – LIP6	6481	1335	0	0,82920	1,00000	0,90662
–	E7.R3 – LIP6	6478	1338	0	0,82881	1,00000	0,90639
3	E6.R2 – IRISA	6464	1352	0	0,82702	1,00000	0,90532
–	E6.R1 – IRISA	6461	1355	0	0,82664	1,00000	0,90509
–	E7.R1 – LIP6	6452	1364	0	0,82549	1,00000	0,90440
–	E6.R3 – IRISA	6449	1367	0	0,82510	1,00000	0,90417
–	E3.R3 – Tweetaneuse	6446	1369	1	0,82482	0,99984	0,90394
4	E5.R3 – Synapse IRIT	6443	1373	0	0,82433	1,00000	0,90371
5	E1.R1 – EDF R&D	6432	1384	0	0,82293	1,00000	0,90286
–	E5.R4 – Synapse IRIT	6425	1391	0	0,82203	1,00000	0,90232
–	E5.R1 – Synapse IRIT	6415	1401	0	0,82075	1,00000	0,90155
–	E6.R4 – IRISA	6414	1402	0	0,82062	1,00000	0,90148
–	E1.R2 – EDF R&D	6411	1405	0	0,82024	1,00000	0,90124
–	E5.R2 – Synapse IRIT	6399	1417	0	0,81871	1,00000	0,90032
6	E8.R2 – Eloquant	6362	1453	1	0,81408	0,99984	0,89745
7	E11.R3 – Syllabs	6300	1516	0	0,80604	1,00000	0,89260
–	E11.R2 – Syllabs	6299	1517	0	0,80591	1,00000	0,89253
8	E9.R2 – EPITA	6292	1524	0	0,80502	1,00000	0,89198
–	E3.R1 – Tweetaneuse	6289	1527	0	0,80463	1,00000	0,89174
–	E9.R3 – EPITA	6279	1537	0	0,80335	1,00000	0,89095
–	E9.R4 – EPITA	6279	1537	0	0,80335	1,00000	0,89095
–	E9.R1 – EPITA	6266	1550	0	0,80169	1,00000	0,88993
–	E11.R1 – Syllabs	6251	1565	0	0,79977	1,00000	0,88875
–	E11.R4 – Syllabs	6243	1573	0	0,79875	1,00000	0,88811
–	E9.R5 – EPITA	6238	1578	0	0,79811	1,00000	0,88772
9	E10.R1 – UTTLM2S	6220	1596	0	0,79580	1,00000	0,88629
–	E8.R1 – Eloquant	6202	1613	1	0,79360	0,99984	0,88486
–	E11.R5 – Syllabs	6191	1625	0	0,79209	1,00000	0,88399
–	E5.R5 – Synapse IRIT	6187	1629	0	0,79158	1,00000	0,88367
10	E2.R2 – CLaC	6093	1723	0	0,77955	1,00000	0,87612
–	E10.R2 – UTTLM2S	6067	1749	0	0,77623	1,00000	0,87402
–	E3.R4 – Tweetaneuse	6046	1769	1	0,77364	0,99983	0,87231
–	E10.R3 – UTTLM2S	6015	1801	0	0,76958	1,00000	0,86979
11	E14.R1 – ADVTeam	5511	2305	0	0,70509	1,00000	0,82704
–	E2.R1 – CLaC	4387	3428	1	0,56136	0,99977	0,71900

TABLE 3 – Décomptes de vrais positifs (VP), faux positifs (FP) et faux négatifs (FN), et résultats (micro-mesures) sur la tâche 1 (classification transport/autre) classés par F-mesure décroissante

une représentation tri-dimensionnelle des valeurs de précision, rappel et F-mesure pour chaque soumission de chaque équipe sur la première tâche. La meilleure valeur de F-mesure est mise en évidence par une flèche (équipe 3, deuxième soumission).

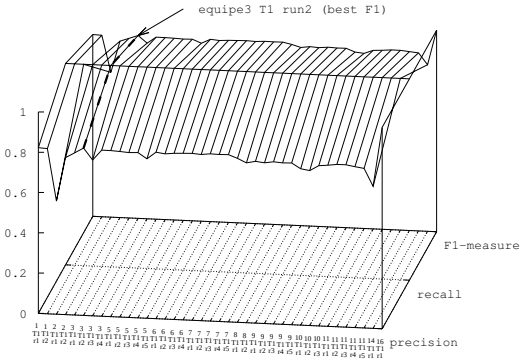


FIGURE 2 – Représentation tri-dimensionnelle des valeurs de précision, rappel et F-mesure pour chaque soumission de chaque équipe sur la première tâche. Le dernier résultat à droite représente la performance des données de référence

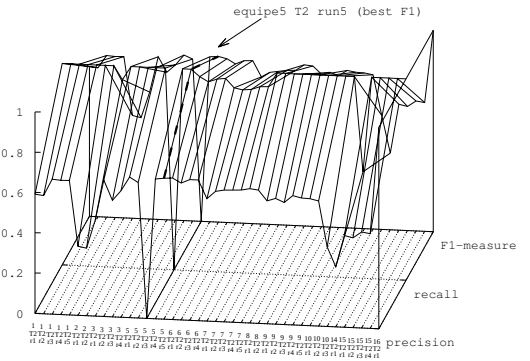


FIGURE 3 – Représentation tri-dimensionnelle des valeurs de précision, rappel et F-mesure pour chaque soumission de chaque équipe sur la deuxième tâche. Le dernier résultat à droite représente la performance des données de référence

4.3 Tâche 2 : polarité globale

Nous reportons dans le tableau 4 les décomptes en termes de vrais positifs, faux positifs et faux négatifs, ainsi que les valeurs de précision, rappel et F-mesure calculées au moyen des micro-mesures sur chacune des soumissions des participants à la tâche 2, classées par F-mesure décroissante. Nous indiquons le rang global de chaque équipe sur la base de la meilleure soumission. La figure 3 fournit une représentation tri-dimensionnelle des valeurs de précision, rappel et F-mesure pour chaque soumission de chaque équipe sur la deuxième tâche. La meilleure valeur de F-mesure est mise en évidence par une flèche (équipe 5, cinquième soumission).

4.4 Significativité statistique

Nous observons une grande homogénéité des résultats, plusieurs équipes obtenant des résultats proches sur plusieurs soumissions. Nous avons alors vérifié les différences entre résultats en mesurant la significativité statistique des différences de résultats. Pour cela, nous avons mis en place le protocole suivant : le jeu de test a été découpé aléatoirement en 30 morceaux et les performances des systèmes sur ces 30 morceaux (valeurs de F-mesure calculées par rapport à la référence) ont été comparées à l'aide d'un t-test pairé et d'un test de Wilcoxon. Le seuil de significativité est fixé à 5 % (p-valeur=0.05).

Les résultats des tests de significativité statistique (t-test) calculés entre soumissions prises deux à deux sont représentés dans les tableaux 5 (première tâche) et 6 (deuxième tâche). Les lignes de ces deux tableaux sont classées par valeurs de micro F-mesure décroissantes, alors que les colonnes sont classées par ordre d'inscription. Ces tableaux se lisent comme suit. La meilleure soumission sur la première tâche (première ligne du tableau 5) est E3.R2 (run 2 de l'équipe Tweetaneuse sachant que E3 correspond à l'équipe Tweetaneuse, cf. tableau 1). Dans la colonne correspondante, on observe des 'o' sur les cinq soumissions suivantes (de E7.R4 à E6.R2), ce qui signifie que les différences observées entre la soumission E3.R2 et les cinq suivantes ne sont pas significatives (p-valeur<0,05). Les différences deviennent significatives (symbole '*') à partir de la soumission E6.R1 (au 7ème

Rang global	Équipe et soumission	Décomptes			Micro-mesures		
		VP	FP	FN	Précision	Rappel	F-mesure
1	E5.R5 – Synapse IRIT	2755	1186	0	0,69906	1,00000	0,82288
–	E5.R1 – Synapse IRIT	2748	1193	0	0,69728	1,00000	0,82165
–	E5.R4 – Synapse IRIT	2734	1207	0	0,69373	1,00000	0,81918
2	E6.R1 – IRISA	2700	1143	98	0,70258	0,96497	0,81313
–	E6.R4 – IRISA	2678	1158	105	0,69812	0,96227	0,80919
3	E3.R2 – Tweetaneuse	2668	1273	0	0,67699	1,00000	0,80738
–	E6.R3 – IRISA	2667	1153	121	0,69817	0,95660	0,80720
–	E5.R2 – Synapse IRIT	2650	1291	0	0,67242	1,00000	0,80413
4	E1.R3 – EDF R&D	2641	1300	0	0,67013	1,00000	0,80249
–	E1.R5 – EDF R&D	2631	1310	0	0,66760	1,00000	0,80067
5	E8.R1 – Eloquant	2628	1313	0	0,66684	1,00000	0,80012
–	E1.R4 – EDF R&D	2625	1316	0	0,66607	1,00000	0,79957
–	E8.R2 – Eloquant	2607	1334	0	0,66151	1,00000	0,79627
–	E6.R2 – IRISA	2569	1212	160	0,67945	0,94137	0,78925
6	E9.R4 – EPITA	2529	1412	0	0,64172	1,00000	0,78176
–	E9.R5 – EPITA	2491	1450	0	0,63207	1,00000	0,77456
7	E10.R1 – UTTLM2S	2483	1458	0	0,63004	1,00000	0,77304
8	E7.R5 – LIP6	2473	1284	184	0,65824	0,93075	0,77113
–	E3.R4 – Tweetaneuse	2473	1467	1	0,62766	0,99960	0,77113
–	E9.R2 – EPITA	2459	1482	0	0,62395	1,00000	0,76844
–	E7.R4 – LIP6	2459	1292	190	0,65556	0,92827	0,76844
–	E10.R2 – UTTLM2S	2446	1495	0	0,62065	1,00000	0,76593
–	E7.R3 – LIP6	2426	1284	231	0,65391	0,91306	0,76205
–	E9.R1 – EPITA	2404	1537	0	0,61000	1,00000	0,75776
–	E7.R2 – LIP6	2390	1311	240	0,64577	0,90875	0,75502
–	E9.R3 – EPITA	2386	1555	0	0,60543	1,00000	0,75423
–	E1.R1 – EDF R&D	2332	1609	0	0,59173	1,00000	0,74350
–	E1.R2 – EDF R&D	2313	1628	0	0,58691	1,00000	0,73969
–	E3.R3 – Tweetaneuse	2279	1662	0	0,57828	1,00000	0,73280
–	E7.R1 – LIP6	2269	1468	204	0,60716	0,91751	0,73076
9	E15.R3 – LIS Lab	1877	2064	0	0,47628	1,00000	0,64524
–	E15.R4 – LIS Lab	1852	2089	0	0,46993	1,00000	0,63939
–	E3.R1 – Tweetaneuse	1814	1997	130	0,47599	0,93313	0,63041
–	E15.R1 – LIS Lab	1793	2148	0	0,45496	1,00000	0,62539
–	E15.R2 – LIS Lab	1755	2186	0	0,44532	1,00000	0,61622
–	E10.R3 – UTTLM2S	1556	2385	0	0,39482	1,00000	0,56613
10	E2.R1 – CLaC	1350	2591	0	0,34255	1,00000	0,51030
–	E2.R2 – CLaC	1320	2621	0	0,33494	1,00000	0,50181
11	E14.R1 – ADVTeam	927	2186	828	0,29778	0,52821	0,38085
–	E5.R3 – Synapse IRIT	0	0	3941	0,00000	0,00000	0,00000

TABLE 4 – Décomptes de vrais positifs (VP), faux positifs (FP) et faux négatifs (FN), et résultats (micro-mesures) sur la tâche 2 (polarité globale) classés par F-mesure décroissante

rang global). De même, les colonnes des soumissions E7.R4 à E6.R2 permettent de constater que les différences ne sont pas significatives. Ainsi émergent des clusters de soumissions, avec un premier cluster composé des runs 2 de Tweetaneuse (E3), des runs 4, 2, 5, 3 du LIP6 (E7) et du run 2 de l'IRISA (E6). Ce phénomène est moins marqué sur la deuxième tâche. Nous observons que seuls les trois premiers runs ne présentent aucune différence du point de vue de nos tests statistiques.

	E1.R1	E1.R2	E3.R1	E3.R2	E3.R3	E3.R4	E5.R1	E5.R2	E5.R3	E5.R4	E5.R5	E6.R1	E6.R2	E6.R3	E6.R4	E7.R1	E7.R2	E7.R3	E7.R4	E7.R5	E8.R1	E8.R2	E9.R1	E9.R2	E9.R3	E9.R4	E9.R5	E11.R1	E11.R2	E11.R3	E11.R4	E11.R5
E3.R2	*	*	*		*	*	*	*	*	*	*	*	o	*	*	*	o	o	o	o	*	*	*	*	*	*	*	*	*	*	*	*
E7.R4	*	*	*	o	*	*	*	*	*	*	*	o	o	*	*	*	o	o		o	*	*	*	*	*	*	*	*	*	*	*	*
E7.R2	*	*	*	o	*	*	*	*	*	*	*	*	o	*	*	*		o	o	o	*	*	*	*	*	*	*	*	*	*	*	*
E7.R5	*	*	*	o	*	*	*	*	o	*	*	o	o	o	*	*	o	o	o		*	*	*	*	*	*	*	*	*	*	*	*
E7.R3	*	*	*	o	*	*	*	*	o	*	*	o	o	o	*	o	o		o	o	*	*	*	*	*	*	*	*	*	*	*	*
E6.R2	*	*	*	o	*	*	*	*	o	*	*	o		o	*	o	o	o	o	o	*	*	*	*	*	*	*	*	*	*	*	*
E6.R1	*	*	*	*	*	*	*	*	o	*	*		o	o	*	o	*	o	o	o	*	*	*	*	*	*	*	*	*	*	*	*
E7.R1	o	o	*	*	*	*	*	*	o	o	*	o	o	o	*		*	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E6.R3	o	*	*	*	*	*	o	*	o	o	*	o	o		*	o	*	o	*	o	*	*	*	*	*	*	*	*	*	*	*	*
E3.R3	o	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	o	*	o	*	o	*	*	o	*
E5.R3	o	*	*	*	*	*	*	*		*	*	o	o	o	*	o	*	o	*	o	*	*	*	*	*	*	*	*	*	*	*	*
E1.R1		o	o	*	o	o	o	o	o	o	o	*	*	o	o	o	*	*	*	*	o	o	o	o	o	o	o	o	o	o	o	o
E5.R4	o	o	*	*	*	*	o	o	*		*	*	*	o	o	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E5.R1	o	o	*	*	*	*		o	*	o	*	*	*	o	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E6.R4	o	o	*	*	*	*	o	o	*	o	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E1.R2	o		*	*	*	*	o	o	*	o	*	*	*	*	o	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E5.R2	o	o	*	*	*	*	o		*	o	*	*	*	*	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E8.R2	o	*	*	*	*	*	*	*	*	*	o	*	*	*	*	*	*	*	*	*	*		*	o	*	o	*	o	*	*	o	o
E11.R3	o	*	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	o	*	o	o	o	*	o		*	*
E11.R2	o	*	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	o	*	o	o	o	*		o	*	*
E9.R2	o	*	o	*	o	o	*	*	*	*	o	*	*	*	*	*	*	*	*	*	*	o	*		*	o	o	o	*	*	o	o
E3.R1	o	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	o	o	o	o	o	o	o	o	o	*
E9.R3	o	*	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	o	*		o	o	o	o	o	o	*
E9.R4	o	*	o	*	o	o	*	*	*	*	o	*	*	*	*	*	*	*	*	*	*	o	o	o	o		o	o	o	o	o	*
E9.R1	o	*	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*	o	o	o	o	o	o	o	*
E11.R1	o	*	o	*	o	o	*	*	*	*	o	*	*	*	*	*	*	*	*	*	*	o	o	o	o	o	o		*	*	o	*
E11.R4	o	*	o	*	o	o	*	*	*	*	o	*	*	*	*	*	*	*	*	*	*	o	o	o	o	o	o	o	*	*		*
E9.R5	o	*	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	o	o	o	o		o	o	o	o	*
E8.R1	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*
E11.R5	o	*	*	*	*	*	*	*	*	*	o	*	*	*	*	*	*	*	*	*	*	o	*	o	*	*	*	*	*	*	*	
E5.R5	o	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	o	*	o	*	o	*	o	*	*	o	o
E3.R4	o	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	o	*	o	*	o	*	*	o	*

TABLE 5 – Différence entre résultats significative (symbole '*', p-value $< 0{,}05$ sur le t-test) et non significative (symbole 'o') sur la tâche 1, dans l'ordre de classement (cf. tableau 3)

	E1.R1	E1.R2	E1.R3	E1.R4	E1.R5	E5.R1	E5.R2	E5.R4	E5.R5	E7.R1	E7.R2	E7.R3	E7.R4	E7.R5	E8.R1	E8.R2	E15.R1	E15.R2	E15.R3	E15.R4
E5.R5	*	*	*	*	*	o	o	o		*	*	*	*	*	*	*	*	*	*	*
E5.R1	*	*	*	*	*		o	o	o	*	*	*	*	*	*	*	*	*	*	*
E5.R4	*	*	*	*	*	o	*		o	*	*	*	*	*	*	*	*	*	*	*
E5.R2	o	o	o	o	o	o		*	o	o	o	o	o	o	o	o	o	o	o	o
E1.R3	o	o		o	o	*	o	*	*	o	o	o	o	o	o	o	o	o	o	o
E1.R5	*	*	o	o		*	o	*	*	*	*	*	*	*	o	o	*	*	*	*
E8.R1	o	o	o	o	o	*	o	*	*	o	o	o	o	o		o	o	o	o	o
E1.R4	*	*	o		o	*	o	*	*	*	*	*	*	*	o	o	*	*	*	*
E8.R2	*	*	o	o	o	*	o	*	*	*	*	*	*	*	o		*	*	*	*
E7.R5	*	*	o	*	*	*	o	*	*	*	*	*	o		o	*	*	*	*	*
E7.R4	*	*	o	*	*	*	o	*	*	*	*	o		o	o	*	*	*	*	*
E7.R3	*	*	o	*	*	*	o	*	*	*	o		o	*	o	*	*	*	*	*
E7.R2	o	*	o	*	*	*	o	*	*	*		o	*	*	o	*	*	*	*	*
E1.R1		o	o	*	*	*	o	*	*	*	o	*	*	*	o	*	*	*	*	*
E1.R2	o		o	*	*	*	o	*	*	o	*	*	*	*	o	*	o	o	o	o
E7.R1	*	o	o	*	*	*	o	*	*		*	*	*	*	o	*	o	o	o	o
E15.R3	*	o	o	*	*	*	o	*	*	o	*	*	*	*	o	*	*	*		*
E15.R4	*	o	o	*	*	*	o	*	*	o	*	*	*	*	o	*	*	*	*	
E15.R1	*	o	o	*	*	*	o	*	*	o	*	*	*	*	o	*		*	*	*
E15.R2	*	o	o	*	*	*	o	*	*	o	*	*	*	*	o	*	*		*	*

TABLE 6 – Différence entre résultats significative (symbole '*', p-value $< 0{,}05$ sur le t-test) et non significative (symbole 'o') sur la tâche 2, dans l'ordre de classement (cf. tableau 4)

4.5 Méthodes des participants

L'utilisation actuelle de méthodes par apprentissage statistique dans les différentes tâches de TAL et la disponibilité de données annotées dans ce défi ont conduit la majorité des participants à se tourner vers des méthodes d'apprentissage supervisé. Les réseaux de neurones convolutifs (CNN) et récurrents (LSTM, biLSTM, et GRU), complétés par des plongements lexicaux, ont ainsi largement été utilisés, tant par les participants d'entreprises tels que EDF R&D (Suignard *et al.*, 2018) ou Synapse Développement & IRIT (Sileo *et al.*, 2018), que par les participants académiques avec les équipes CLaC (Jacques *et al.*, 2018), de l'EPITA (Sainson *et al.*, 2018), de l'IRISA (Minard *et al.*, 2018), du LIP6 (Dias *et al.*, 2018), du LIRMM (Azmy *et al.*, 2018), du LIS (Htait, 2018) et Tweetaneuse (Buscaldi *et al.*, 2018). Les algorithmes d'apprentissage supervisé traditionnels (arbres de décision, bayésien naïf, entropie maximale, CRF, SVM) ont été plus rarement utilisés, parfois à titre de comparaison avec d'autres méthodes, par l'entreprise Syllabs (Monnin *et al.*, 2018) et par les équipes CLaC (Jacques *et al.*, 2018), IRISA (Minard *et al.*, 2018) et Tweetaneuse (Buscaldi *et al.*, 2018). L'entreprise Eloquant (Graceffa *et al.*, 2018) a employé des méthodes symboliques, en adaptant aux spécificités du défi la méthode employée en interne pour traiter des données de relations clients. Cette adaptation passe notamment par un enrichissement sémantique et une prise en compte des propriétés de surface des messages postés sur les réseaux sociaux.

5 Conclusion

L'édition 2018 du défi fouille de texte (DEFT) s'est révélée un succès en terme de nombre de participants, portée par une thématique connue mais toujours en vogue (la fouille d'opinion) et un ensemble de techniques (classification, notamment par réseaux de neurones récurrents) faciles à mettre-en-œuvre. Parmi les quatre tâches proposées, seules les deux premières ont été traitées par les participants. Sur l'identification de la thématique des transports, la micro F-mesure varie de 0,827 à 0,908 tandis que pour l'identification de la polarité globale, la micro F-mesure varie de 0,381 à 0,823.

Nous observons des résultats très homogènes entre participants sur chacune des deux premières tâches, que nous pouvons expliquer par plusieurs points :

— les techniques utilisées sont similaires ;
— la première tâche était relativement facile, probablement parce que la sélection initiale des tweets s'est faite sur mots-clés, amenant les systèmes à des performances quasi optimales ;
— la deuxième tâche, plus difficile au vu des résultats qui doivent être relativisés par le bruit résiduel présent dans les annotations, a aussi amené les systèmes à une sorte de plafond, difficile à dépasser au regard des données.

Le faible nombre de participants (une seule équipe) sur la tâche 3 et l'absence de participants sur la tâche exploratoire est un peu décevant. Bien que nécessitant des techniques d'annotations de texte plutôt que de classification comme sur les deux premières tâches, de nombreux outils et retours d'expérience sur des tâches similaires sont disponibles et auraient pu permettre à un plus grand nombre de participer. Le calendrier serré de cette édition (au maximum un mois et demi pour la phase d'entraînement) aura pu décourager certaines équipes de s'engager dans ces tâches plus complexes.

Remerciements

Le corpus de la campagne d'évaluation DEFT2018 a été produit dans le cadre du projet REQUEST (Programme d'Investissement d'Avenir, appel Cloud computing & Big Data, convention 018062-25005) et annoté en collaboration avec ELDA. Le projet MIROR, du programme de recherche et d'innovation de l'Union Européenne Horizon 2020, « Marie Sklodowska-Curie grant agreement No 676207 », a contribué à l'organisation de la campagne DEFT 2018.

Références

AZMY W. M., MOULAHI B., BRINGAY S., AZÉ J. & SERVAJEAN M. (2018). Lirmm@deft-2018 – modèle de classification de la vectorisation des documents. In *Actes de DEFT*, Rennes, France.

BUSCALDI D., LE ROUX J. & LEJEUNE G. (2018). Modèles en caractères pour la détection de polarité dans les tweets. In *Actes de DEFT*, Rennes, France.

DIAS C.-E., GAINON DE FORSAN DE GABRIAC C., GUIGUE V. & GALLINARI P. (2018). DEFT 2018 : Attention sélective pour classification de microblogs. In *Actes de DEFT*, Rennes, France.

FRAISSE A. & PAROUBEK P. (2014). Toward a unifying model for opinion, sentiment and emotion information extraction. In *Proc of LREC*, p. 3881–3886, Reykjavik, Iceland.

GRACEFFA D., RAMOND A., DUSSERRE E., KALITVIANSKI R., RUHLMANN M. & PADRÓ M. (2018). Notre tweet première fois au DEFT-2018 : systèmes de détection de polarité et de transports. In *Actes de DEFT*, Rennes, France.

HAMON T., FRAISSE A., PAROUBEK P., ZWEIGENBAUM P. & GROUIN C. (2015). Analyse des émotions, sentiments et opinions exprimés dans les tweets : présentation et résultats de l'édition 2015 du défi fouille de texte (DEFT). In *Actes de DEFT*, Caen, France.

HTAIT A. (2018). Adapted sentiment similarity seed words for french tweets' polarity classification. In *Actes de DEFT*, Rennes, France.

JACQUES S., FARAHNAK F. & KOSSEIM L. (2018). CLaC @ DEFT 2018 : Sentiment analysis of tweets on transport from île-de-France. In *Actes de DEFT*, Rennes, France.

MANNING C. D. & SCHÜTZE H. (2000). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts : MIT Press.

MINARD A.-L., RAYMOND C. & CLAVEAU V. (2018). Participation de l'IRISA à DeFT 2018 : classification et annotation d'opinion dans des tweets. In *Actes de DEFT*, Rennes, France.

MONNIN C., QUERNÉ O. & HAMON O. (2018). Syllabs@DEFT2018 : combinaison de méthodes de classification supervisées. In *Actes de DEFT*, Rennes, France.

SAINSON A., LINSENMAIER H., MAJED A., CADET X. & BOUCHEKIF A. (2018). LSE au DEFT 2018 : Classification de tweets basée sur les réseaux de neurones profonds. In *Actes de DEFT*, Rennes, France.

SILEO D., VAN DE CRUYS T., MULLER P. & PRADEL C. (2018). Concaténation de réseaux de neurones pour la classification de tweets, DEFT2018. In *Actes de DEFT*, Rennes, France.

SUIGNARD P., CHARAUDEAU L., BOUMGHAR M., BOTHUA M. & LAGARDE D. (2018). Participation d'EDF R&D à DEFT 2018. In *Actes de DEFT*, Rennes, France.

Participation d'EDF R&D à DEFT 2018

Philippe Suignard, Lou Charaudeau, Manel Boumghar, Meryl Bothua, Delphine Lagarde
EDF R&D, 7 boulevard Gaspard Monge, 91120 Palaiseau
prenom.nom@edf.fr

RESUME

Ce papier décrit la participation d'EDF R&D à la campagne d'évaluation DEFT 2018. Notre équipe a participé aux deux premières tâches : classification des tweets en transport/non-transport (Tâche T1) et détection de la polarité globale des tweets (Tâche T2). Nous avons utilisé 3 méthodes différentes s'appuyant sur Word2Vec, CNN et LSTM. Aucune donnée supplémentaire, autre que les données d'apprentissage, n'a été utilisée. Notre équipe obtient des résultats très corrects et se classe 1ère équipe non académique. Les méthodes proposées sont facilement transposables à d'autres tâches de classification de textes courts et peuvent intéresser plusieurs entités du groupe EDF.

ABSTRACT

Here the title in English.
This paper describes the participation of EDF R&D at DEFT 2018 evaluation campaign. Our team participated in the first two tasks: classification of tweets in transport / non-transport (Task T1) and detection of the overall polarity of tweets (Task T2). We used 3 different methods based on Word2Vec, CNN and LSTM. No additional data other than the training data was used. Our team gets very correct results and ranks 1st non-academic team. The proposed methods are easily transferable to other short text classification tasks and may interest several entities of the EDF group.

MOTS-CLÉS : Tweet, Polarité, Word2Vec, LSTM, CNN.
KEYWORDS: Tweet, Polarity, Word2Vec, LSTM, CNN.

1 Introduction

Plusieurs éléments nous ont motivés à participer à l'édition 2018 du défi DEFT (Paroubek, 2018) :

- EDF R&D travaille en appui de la Direction EDF Commerce à la mise en œuvre d'une chaine de récupération des tweets, de classification et d'attribution d'une polarité (travail en cours qui s'apparente à la tâche 2).
- Dans la phase de récupération des tweets, nous avons une problématique similaire à la tâche 1, non pas sur la distinction des tweets « transport » / « inconnu », mais EDF « Electricité de France » / EDF « Equipe de France ».
- La volumétrie importante des tweets mis à disposition pour ce concours nous permettait d'envisager des méthodes de type Machine Learning.

Participer à DEFT était l'occasion de travailler sur plusieurs méthodes de classification dont les résultats contribueront directement à EDF Commerce et à d'autres entités du groupe EDF.

2 Description des méthodes utilisées

Les trois méthodes que nous avons proposées partagent la même approche à savoir d'entrainer deux modèles chacune (un pour la tâche 1 et un autre pour la tâche 2). Le premier modèle permet de discriminer les tweets entre les catégories « transport » et « inconnu ». Le second modèle discrimine les tweets selon leur polarité.

Deux corpus étaient fournis pour l'apprentissage « batch_b », composé d'environ 14 000 tweets, et « simple », composé d'environ 54 000 tweets. Comme tous les tweets de « batch_b » étaient étiquetés « inconnu », nous avons préféré ne pas les utiliser.

Aucune donnée externe supplémentaire n'a été utilisée. Pour entrainer nos méthodes, nous avons découpé le corpus « simple » en deux parties : 80% étant réservé pour l'apprentissage et 20% ont servi pour les tests. Quelques tweets vides ou erronés ont été éliminés.

2.1 Méthode 1 basée sur Word2Vec

La méthode 1 est inspirée de (Xing, 2017), elle consiste à entrainer un modèle Word2Vec à la fois sur les mots des tweets et sur leur catégorie. A l'aide du modèle Word2Vec ainsi entrainé, une série de descripteurs sont calculés pour chaque tweet pour ensuite entrainer un classifieur. La suite de ce paragraphe présente les différentes étapes suivies.

2.1.1 Nettoyage des tweets

Les tweets sont pré-traités de la manière suivante :

- Passage des mots en minuscule ;
- Suppression des lettres redoublées, ce qui transforme « cooooooool » en « col », « arrêt » en « arêt », « mdrrrr » et « mddddrrr » en « mdr ». Cela permet de réduire le vocabulaire et de s'affranchir d'une certaine partie des fautes d'orthographe ;
- Transformation des URL par « HTTP » ;
- Changement des dates du type 01/02/2016 par « DATE » ;
- Changement des heures du type 12h30 ou 12:20 par « HEURES » ;
- Changement des durées du type 12 min, 12mn, 12 minutes par « DUREE » ;
- Transformation des smiley par « SMILEYHAPPY », « SMILEYSAD », « SMILEYWINK », « SMILEYCRYING », etc. sans toutefois être exhaustif ;
- Suppression des caractères de mention et de hashtag : « @ratp » devient « ratp » et « #sncf » devient « sncf ».

2.1.2 Word2Vec

On ne présente plus la méthode Word2Vec (Mikolov, 2013) qui consiste à transformer des mots en vecteurs.

La 1ère méthode pour classifier les tweets va s'appuyer sur Word2Vec, avec comme idée centrale, le fait d'entrainer un modèle en mélangeant les mots d'un tweet avec la catégorie de ce tweet. Pour entrainer le modèle pour la tâche 1, le tweet « Les gars qui puent des aisselles dans le bus c'est vous » ayant pour catégorie « TRANSPORT », devient :

– <transport> Les gars qui puent des aisselles dans le bus c'est vous <transport>

Pour entrainer le modèle pour la tâche 2, il devient :

– <negatif> Les gars qui puent des aisselles dans le bus c'est vous <negatif>

Ainsi, un mot fréquemment utilisé dans la catégorie « transport » et très connoté « transport » aura tendance à être d'avantage similaire (au sens de Word2Vec) du mot « transport ». Il en va de même pour les catégories « inconnue », « positif », « négatif », « neutre » et « mixte ».

Les paramètres retenus pour entrainer les modèles Word2Vec sont : Skip-Gram, un voisinage de 5 mots à droite et à gauche, une couche cachée de taille 200, un softmax hiérarchique, une fréquence minimale des mots de 30, les mots de un ou deux caractères sont éliminés et 1000 itérations pour entrainer le modèle.

Une stop-liste a été utilisée pour éliminer les mots ayant peu de valeur ajoutée pour la problématique concernée.

2.1.3 Calcul des descripteurs

Pour chaque tweet du corpus d'entrainement, sont calculés les descripteurs suivants :

- Similarité **vectorielle moyenne** : calcul de la moyenne des vecteurs mots du tweet et calcul de la similarité entre ce vecteur mot et les vecteurs mots des catégories « transport » et « inconnu » pour la tâche 1 (soit 2 descripteurs pour T1 et 4 pour T2).
- Similarité **distributionnelle** : calcul de la distribution des N plus proches voisins des mots du tweet (au sens de Word2Vec), calcul de la distribution des N plus proches voisins des catégories « transport » et « inconnu » pour la tâche 1, puis calcul de la similarité entre ces deux distributions. Ce qui fait 2 descripteurs pour T1 et 4 pour T2.
- Similarité **centrale** : calcul préalable des vecteurs moyens pour chaque catégorie du corpus d'apprentissage (un vecteur pour la catégorie « transport » et un autre pour « inconnu », par exemple), puis calcul de la similarité entre chaque tweet et ces tweets moyens, soit 2 descripteurs pour T1 et 4 pour T2.
- **Moyenne des similarités des mots** : calcul de la similarité entre chaque mot du tweet et les différentes catégories (« transport » et « inconnu » pour la tâche 1), puis calcul de la moyenne de ces similarités, soit 2 descripteurs pour T1 et 4 pour T2.
- **Maximum des similarités des mots** : même chose que précédemment, mais en retenant uniquement le maximum des similarités, soit 2 descripteurs pour T1 et 4 pour T2.
- **Minimum des similarités des mots** : même chose que précédemment, mais en retenant uniquement le minimum des similarités, soit 2 descripteurs pour T1 et 4 pour T2.
- **Ecart types des similarités des mots** : même chose que précédemment, mais en retenant uniquement les écarts types des similarités, soit 2 descripteurs pour T1 et 4 pour T2.

Pour la tâche 2, comme on cherche à détecter la présence de mots positifs et de mots négatifs au sein d'un même tweets, notamment pour détecter les tweets de la catégorie « mixte », on ajoute les descripteurs suivants :

- **Maximum des ruptures de similarité entre deux mots successifs** ;
- **Minimum des ruptures de similarité entre deux mots successifs** ;
- **Maximum des ruptures de similarité entre deux mots du tweet** ;
- **Minimum des ruptures de similarité entre deux mots du tweet**, soit 4x4 = 16 descripteurs supplémentaires pour T2;

Ce qui fait au total 14 descripteurs par tweet pour la tâche T1 et 44 pour la tâche 2.

2.1.4 Entrainement d'un classifieur

Les descripteurs ont été calculés pour chaque tweet du corpus d'apprentissage puis convertis au format ARFF pour être utilisées au sein du logiciel WEKA (Hall, 2009). Plusieurs classifieurs ont été testés : « Régression Logistique » et « Random Forest ».

Les classifieurs ainsi entrainés ont été appliqués sur les 7816 tweets du corpus d'évaluation. Un seul « run » a été proposé pour la tâche 1 (avec Random Forest, l'écart entre les deux méthodes n'étant pas significatif) et deux « run » ont été proposés pour la tâche 2, respectivement avec les méthodes « Régression Logistique » et « Random Forest ».

2.2 Méthode 2 : LSTM + prétraitement

2.2.1 Annotation des tweets pour la détection de polarité

Nous avons ajouté automatiquement des étiquettes de polarité pour aider l'apprentissage des modèles LSTM et CNN lors de la tâche 2. Nous avons ainsi constitué manuellement deux lexiques, l'un contenant des mots de polarité positive, l'autre contenant des mots de polarité négative. Ces mots sont issus du corpus de tweets. Notre lexique de polarité positive comprend 121 entrées et celui de polarité négative 291. Ils intègrent tous deux des émoticônes. Ils ne prétendent pas à l'exhaustivité mais les annotations qui en découlent ajoutent des métadonnées pour les modèles LSTM et CNN. Ces annotations sous la forme d'étiquettes <POSITIF> ou <NEGATIF> sont ajoutées automatiquement dans les tweets qui comprennent un mot appartenant à l'un ou l'autre des lexiques. Ces prétraitements ont été appliqués sur le corpus d'entraînement et sur le corpus de test. En voici deux exemples :

> 699945537311793152 préavis de grève <**NEGATIF**> à la @sncf pour ce jeudi en rhônealpes et jusqu au 23 février. risques de perturbations <**NEGATIF**> sur le trafic <**NEGATIF**> ter

> 715780410165301248 j suis trop bien <**POSITIF**> dans le bus aek ma musique et le chaufage à côté ☺ **<POSITIF>**

Cet ajout de polarité nous a fait gagner environ 5% en « accuracy » sur le corpus d'apprentissage.

2.2.2 Entrainement de 2 LSTM

Pour cette deuxième méthode, nous avons choisi d'utiliser des réseaux de neurones, basés sur des LSTM (Hochreiter, 1997), qui obtiennent de bonnes performances sur des tâches de classification textuelle (Wenpeng, 2017). Les LSTM étant plus performants lorsque la classification à réaliser est concentrée sur une seule thématique, nous avons choisi d'entrainer un réseau par tâche plutôt qu'un seul global. Nous avons ainsi entrainé 2 réseaux de neurones séparés pour chaque tâche de classification : le premier avait pour objectif de séparer les tweets entre « INCONNU » et « TRANSPORT » et le second d'attribuer une polarité aux tweets identifiés dans la catégorie « TRANSPORT »

Chaque réseau a été construit en associant une première couche d'embedding, un LSTM puis une couche dense en sortie de manière à combiner les sorties du LSTM. Pour chaque tâche, nous avons réalisé une exploration des paramètres suivants :

- Taille du padding des phrases d'entrée ;
- Taille d'embedding ;
- Taille de la couche cachée du LSTM.

Pour chaque paramètre, nous avons sélectionné une plage de variation, puis entrainé 10 modèles sur chaque jeu de paramètres différents.

Les performances moyennes des modèles ont été estimées pour chaque jeu de paramètres selon 2 critères :

- **L'accuracy du modèle sur le jeu de validation** : pourcentage de bonnes classifications réalisées par le modèle après entrainement
- **La « loss » sur le jeu de validation** : somme des erreurs réalisées par le modèle sur le jeu de validation

Pour la première tâche, chaque modèle a été entrainé sur une sélection aléatoire de 80% des tweets du jeu de tweets d'entrainement puis validé sur les 20% restants. Pour la seconde tâche, les modèles ont été entrainés sur 80% des tweets étiquetés « TRANSPORT » puis validés sur les 20% restants. L'apprentissage a été réalisé sur des mini-batch de 32 échantillons.

Les meilleures performances ont été obtenues pour les paramètres suivants :

- Première tâche : embedding de taille 150, padding de taille 40 et couche cachée du LSTM de 60 neurones
- Deuxième tâche : embedding de taille 300, padding de taille 40 et couche cachée du LSTM de taille 100

Si le choix de paramètres pour la première tâche est relativement classique, la tâche de classification étant dans un cadre simple, il est plus complexe pour la deuxième tâche : la catégorisation plus complexe pousse à augmenter le nombre de neurones du réseau, alors que le nombre de données d'apprentissage est plus réduit, ce qui augmente le risque de surapprentissage. Nous avons ainsi choisi d'orienter les modèles vers des réseaux plus grands, tout en ajoutant un dropout de 20% pour contrer le surapprentissage.

2.3 Méthode 3 : CNN + prétraitement

2.3.1 Annotation des tweets pour la détection de polarité

Nous avons appliqué les mêmes prétraitements que ceux décrits en 2.2.1.

2.3.2 Entrainement du CNN

Enfin, pour la troisième méthode, nous avons choisi d'utiliser des réseaux de neurones, basés sur des CNN (LeCun, 1998). Le caractère hiérarchique des CNN en font de bons candidats pour traiter des tâches de classification de textes. En effet, ces structures ont été fréquemment utilisées dans la littérature pour des tâches de classification notamment d'analyse de sentiment (Dauphin, 2016).

Les CNN ont été entraînés uniquement pour la deuxième tâche, c'est-à-dire l'attribution d'une polarité aux tweets identifiés dans la catégorie « TRANSPORT ».

Pour cette tâche nous avons choisi d'explorer les paramètres ci-dessous :

- Taille du padding des phrases d'entrée
- Taille d'embedding
- Taille de la fenêtre de filtre
- Taille de la fenêtre de pooling

De la même manière que pour les LSTM, les performances moyennes des modèles ont été estimées pour chaque jeu de paramètres selon 2 critères :

- L'accuracy du modèle sur le jeu de validation
- La loss sur le jeu de validation

Aussi, pour l'entrainement des modèles, ces derniers ont été entrainés sur 80% des tweets étiquetés « TRANSPORT » puis validés sur les 20% restants. L'apprentissage a été réalisé sur des mini-batch de 32 échantillons.

Les meilleures performances ont été obtenues pour les paramètres suivants :

- Paramètres retenus pour la deuxième tâche : embedding de taille 300, padding de taille 40 et une taille de filtre et de pooling de 5 mots

3 Résultats obtenus

Les tables 1 et 2 récapitulent les résultats obtenus par nos méthodes sur les tâches 1 et 2. F-Mesure est la F-Mesure obtenu par nos méthodes, F-Mesure moyenne, est la moyenne des F-Mesure de tous les run de tous les participants et F-Mesure Max est le maximum des F-Mesures des participants.

Tâche 1, nom de la méthode :	F-Mesure	F-Mesure moyenne	F-Mesure Max
Word2Vec + Classifieur Random Forest	**0,90286**	0,88813	0,90785
LSTM + Prétraitements	0,90124	0,88813	0,90785

TABLE 1 – Résultats de notre participation à la campagne Deft 2018 pour la tâche 1.

Tâche 2, nom de la méthode :	F-Mesure	F-Mesure moyenne	F-Mesure Max
Word2Vec + Classifieur Regression Logistique	0,7435	0,7293	0,82288
Word2Vec + Classifieur Random Forest	0,73969	0,7293	0,82288
LSTM + Prétraitements	**0,80249**	0,7293	0,82288
LSTM2(*) + Prétraitements	0,79957	0,7293	0,82288
CNN + Prétraitements	0,80067	0,7293	0,82288

TABLE 2 – Résultats de notre participation à la campagne Deft 2018 pour la tâche 2.

(*) : LSTM2 est la même méthode que LSTM mais en restreignant la taille du vocabulaire.

De ces résultats, nous tirons les enseignements suivants :

- EDF R&D se place 5ème sur la tâche 1 (sur 11 équipe) et 4ème sur la tâche 2 (sur 12 équipes).
- Tous nos « run » ont une « F-mesure » au-dessus de la moyenne des F mesures.
- Nous sommes classés 1er acteur non académique.
- Sur la tâche 1, c'est la méthode basée sur Word2Vec qui obtient, de très peu, le meilleur résultat.
- Sur la tâche 2, c'est la méthode basée sur les LSTM qui obtient les meilleurs résultats.

Selon notre point de vue, notre participation à ce concours est positive, car elle nous a permis de tester plusieurs méthodes de classification de textes courts.

4 Conclusion

Participer à la campagne DEFT 2018, nous a permis de tester 3 méthodes de classifications de textes courts basées sur Word2Vec, LSTM et CNN. Aucune donnée supplémentaire, autre que les données d'apprentissage, n'a été utilisée. Les résultats obtenus sont satisfaisants. Les méthodes que nous avons mises en œuvre sont facilement transposables à d'autres tâches de classification de textes courts et peuvent intéresser plusieurs entités du groupe EDF.

Références

DAUPHIN Y. N., FAN, A., AULI M., & GRANGIER D. (2016). Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*.

HALL M., FRANK E., HOLMES G., PFAHRINGER B., REUTEMANN P., & WITTEN I.H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

HOCHREITER .S, AND SCHMIDHUBER J. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

LECUN Y., BOTTOU L., BENGIO Y., & HAFFNER P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S., & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

MIKOLOV T., CHEN K., CORRADO G., & DEAN J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L., TORRES-MORENO J.M. DEFT2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en Île de France. In: Actes de DEFT. Rennes, France.

WENPENG Y., KATHARINA K., MO Y., HINRICH S. (2017) Comparative study of CNN and RNN for language processing. ARXIV PREPRINT ARXIV :1702.01923.

XING L., & PAUL M. J. (2017). Incorporating Metadata into Content-Based User Embeddings. In Proceedings of the 3rd Workshop on Noisy User-generated Text (pp. 45-49).

CLaC @ DEFT 2018: Sentiment analysis of tweets on transport from Île-de-France

Simon Jacques Farhood Farahnak Leila Kosseim
Dept. of Computer Science and Software Engineering
Concordia University
1455 De Maisonneuve Blvd. W. Montreal, Canada
s-jacques@live.com, farhood.farahnak@gmail.com,
leila.kosseim@concordia.ca

RÉSUMÉ

Analyse de tweets sur les transport sur l'Île-de-France

Cet article décrit le system développé par le laboratoire CLaC de l'Université Concordia à Montréal pour la campagne DEFT 2018. La compétition comptait quatre tâches differentes, parmis lesquelles nous avons participé aux deux premières. Nous avons utilisé deux méthodes d'apprentissage supervisé: une machine à vecteurs de support et un réseau de neuronnes. À la tâche 1, notre mesure-F la plus élevée atteint 87.61% et à la tâche 2, elle atteint 51.03%, situant notre système en dessous de la moyenne par rapport aux autres participants.

ABSTRACT

CLaC @ DEFT 2018: Analysis of tweets on transport on the Île-de-France

This paper describes the system deployed by the CLaC lab at Concordia University in Montreal for the DEFT 2018 shared task. The competition consisted in four different tasks; however, due to lack of time, we only participated in the first two. We participated with a system based on conventional supervised learning methods: a support vector machine classifier and an artificial neural network. For task 1, our best approach achieved an F-measure of 87.61%; while at task 2, we achieve 51.03%, situating our system below the average of the other participants.

MOTS-CLÉS : Machine à vecteurs de support; Analyse de sujets; Anlyse de sentiments.

KEYWORDS: Support Vector Machine; Topic Analysis; Sentiment Analysis.

1 Introduction

This paper describes the system deployed by the CLaC Lab at Concordia University for the DEFT 2018 shared task. For this 14^{th} edition of the *Défi Fouille de Textes* (DEFT), the main goal was to analyze the sentiments in French tweets regarding transport on the Île-de-France. As described in (Paroubek et al., 2018), four tasks were proposed:

task 1 (T1) – Transport / non-transport classification: Given a tweet, determine whether it concerns transport or not.

task 2 (T2) – Global polarity: Given a transport tweet, determine its overall polarity, chosen from 4 potential classes: positive, negative, neutral or mixed (mixposneg).

Task 3 (T3) – Sentiment marker and target: Given a tweet about transport that expresses at least one sentiment, for each sentiment expressed, determine (1) the target of the sentiment, and (2) the sentiment marker (i.e. the linguistic expression that expresses the sentiment).

Task 4 (T4) – Full annotation: Given a tweet about transport that expresses at least one sentiment, for each sentiment expressed, determine (1) the target of the sentiment, (2) the sentiment marker, in addition to (3) the source of the sentiment.

As each task builds on top of the previous one, due to lack of time, we only participated in the first two tasks (T1 and T2).

2 Datasets

The DEFT 2018 organizers (Paroubek et al., 2018) put at our disposal a training corpus of 68,916 tweets already annotated with their topic label (*transport/non-transport*) and their polarity label (*positive, negative, neutral, mixed*). To train our classifiers, we first split the dataset randomly to create two distinct sub-sets: (i) 80% was used as training set, (ii) and 20% was used as validation set. Since the polarity was identified only on transport tweets, the second task consisted of 28,374 tweets for training and 7,094 for validation. Tables 1 and 2 show the distribution of the datasets for tasks T1 and T2, respectively.

Table 1: Training and validation sets for task 1

Label	**Training**	**Validation**	**Total**	**Proportion**
Transport	28,374	7,093	35,468	51.47%
Non-Transport	26,758	6,690	33,448	48.53%
Total	55,132	13,783	68,916	100%

Table 2: Training and validation sets for task 2

Label	**Training**	**Validation**	**Total**	**Proportion**
Neutral	10,089	2,522	12,611	35.56%
Positive	5,862	1,466	7,328	20.66%
Negative	10,487	2,622	13,109	36.96%
Mixposneg	1,936	484	2,420	6.82%
Total	28,374	7,093	35,468	100%

3 Pre-Processing

Given the original datasets (see Section 2), we first performed various pre-processing steps to clean and normalize the tweets. These steps were inspired by the work of (Mohammad et al., 2013; Pak & Paroubek, 2010; Reitan et al., 2015).

1. **UTF-8 Encoding**: To facilitate processing, all messages in the corpus were encoded with the format *UTF-8*.

Step	Message
1. UTF-8 Encoding	"#EURO2016 #ALLFRA : à #Lille , même les bus supportent les #Bleus. C' est le Noooord https://t.co/ws5s9LyiR7 https://t.co/HEX9ksiPSH"
2. Hypertext Removal	"#EURO2016 #ALLFRA : à #Lille , même les bus supportent les #Bleus. C' est le Noooord"
3. Case folding	"#euro2016 #allfra : à #lille , même les bus supportent les #bleus. c' est le noooord"
4. Special Character Removal	"euro2016 allfra à lille même les bus supportent les bleus. c' est le noooord"
5. Character Repetition Reduction	"euro2016 allfra à lille même les bus supportent les bleus. c' est le noord"
6. Tokenization	euro2016, allfra, à, lille, même, les, bus, supportent, les, bleus, ., c', est, le, noord
7. Stopword Removal	euro2016, allfra, lille, bus, supportent, bleus, ., noord

Figure 1: Example of pre-processing of tweets

2. **Hyperlink Removal**: All links starting with *http*, *https*, or *www* were removed from the tweets.

3. **Case Folding**: All uppercase characters were converted to lowercase.

4. **Special Character Removal**: To limit the character set, we only considered 45 possible characters. This 45-character set was composed of all 26 French letters, plus 12 with diacritics, and 7 punctuation marks including the hyphen. All characters not included in this predefined set, in particular hashtags (#), were removed from the tweets. This allowed us to focus the classification on words.

5. **Character Repetition Reduction**: All words that included more than two consecutive identical characters were reduced to only two consecutive characters. For example, as shown in Figure 1, *noooord* was reduced to *noord*.

6. **Punctuation Removal**: For the task 1, we removed all punctuation marks; however, as punctuation has been shown to signal sentiment (Mohammad et al., 2013), we kept them for task 2.

7. **Tokenization**: Once the characters were pre-processed, we tokenized the filtered tweets. For this, we used the French version of `word_tokenize` from the NLTK Toolkit (Loper & Bird, 2002).

8. **Stopword Removal**: We used a list of 156 stopwords to further filter the tweets. 130 stopwords came from the NLTK Tookkit (**?**), and the remaining 26 were added following a manual corpus analysis of the word distribution in the DEFT-2018 dataset.

Figure 1 illustrates the pre-processing of a sample tweet. As shown in Table 3, after pre-processing, the size of tweets was reduced to almost half their size for each label and for both tasks.

As part of the pre-processing, we also experimented with marking negation, in order to increase our performance on task T2. As shown by several previous work(e.g. (Reitan et al., 2015; Kouloumpis

Table 3: Average tweet size before and after pre-processing

Task	Label	Average Nb Words	
		Before Pre-processing	**After Pre-processing**
T1	Transport	22.13	12.17
	Non-Transport	22.62	12.59
T2	Neutral	21.36	11.76
	Positive	22.10	12.07
	Negative	22.58	12.47
	Mixposneg	23.82	13.10

et al., 2011), negation is an important feature for sentiment analysis on tweets. We therefore tried to mark the scope of negation by marking each expression indicating a negation (e.g. *pas, ne, n'*) until the next punctuation. Unfortunately, our elementary method to mark negation seemed to lower the performance with our baseline model, a Naive Bayes Classifier (see Section 4.2) on the the validation set. With the negation marking, the F-measure scored approximately 15% lower than without it. Hence, we dropped our negation marking in the pre-processsing.

4 Experiments

4.1 Features and Feature Selection

We experimented with two types of features and two feature selection methods for a total of 4 experiments. As features, we used (1) Words as feature with binary bag-of-words and frequency bag-of-words representation. (2) as character n-grams have successfully been used on tweets in previous work (e.g. (Reitan et al., 2015)), we also considered character n-grams as features with frequency. We experimented with 3-grams, 4-grams, and 5-grams. While the 5-gram model was too large to be trained efficiently, the 3-gram and 4-gram models were later dropped due to their low performance on the validation set.

As for feature selection, we experimented with 2 simple methods: (1) removing low-frequency features ($<$ *s*ome value n), and (2) removing features with a low entropy difference ($\leq$ some value *t*) between the classes.

4.2 Models

We experimented with 4 different classifiers for task 1, and 3 classifiers for task 2.

1. **Naive Bayes Classifier**: As a baseline, we trained a Naive Bayes Classifier from the NLTK library. We used words as features and Boolean values indicating their presence or absence in the tweet as feature values. This lead to an F-measure of 79.91% on the validation set for task 1; but on task 2, however, the F-measure dropped to 64.18%, which we attempted to improve by using other models.

Table 4: Confusion matrix of the ANN for task 1 on the validation set

Actual \ Predicted	Transport	Non-Transport	Total
Transport	4340	203	4543
Non-Transport	1264	3060	4324
Total	5604	3263	8867
F-Measure	85.54%		

2. **Decision Tree Classifier**: The second model we trained was a Decision Tree Classifier, also from the NLTK library, using the same features vectors as the Naive Bayes classifier. This model lead to an F-measure of 69.97% for task 1 and 57.20% for task 2 on the validation set. It scored much lower than the Naive Bayes classifier, which lead us to drop it entirely for the official runs.

3. **Support Vector Machine**: The third model we trained was a Support Vector Machine classifier from the *scikit-learn* library (Pedregosa et al., 2011). Four versions of this model were experimented with: two models using word-feature, and two models using character n-grams.

   For task 1, we used four different approaches to train our model. When we used words as features and filtered out words with frequency < 4 (i.e. $n = 4$, see Section 4.1), we reached an F-measure of 83.00% on the validation set. On the other hand, when filtering features with an entropy difference $= 0.25$ (see Section 4.1), we reached an F-measure of 83.05% on the validation set. We then trained with character 3-grams and 4-grams, but only reached F-measures of 69.14% and 78.16% respectively. Because the n-gram models achieved a lower performance than the word-based models, we did not use them for the actual shared task (see Section 5).

   For task 2, due to lack of time, we only experimented with two different approaches to train our model. The first one was based on words with frequency $= 4$ as features and an entropy difference $= 0.25$. This model reached an F-measure of 68.47% on the validation set. The second model used character 4-grams as features, since they seemed to achieve a higher performance than 3-grams on task 1. The 4-gram only managed to reach an F-measure of 42.72%, significantly lower than the word-based model. Again, the n-gram models were therefore discarded.

   Table 5 shows the confusion matrix of the best validation for this model on task 2.

4. **Artificial Neural Network**: The last model we trained was an classic Neural Network. We used words with frequency $= 4$ as features and an entropy difference $= 0.25$. Our model used a binary bag-of-words representation at the input layer, used two layers with the ReLU activation function (Nair & Hinton, 2010) and trained with the RMSprob optimization algorithm (Tieleman & Hinton, 2012). We also applied dropout (Srivastava et al., 2014) after each layer to prevent over-fitting. This model achieved an F-measure of 85.54% on the validation set.

Following the results of our experiments with the validation set, the best models seemed to be the ANN for task 1, and the SVM with the binary bag-of-words filtered with $n = 4$ and $t = 0.25$ for task 2. Tables 4 and shows the confusion matrix of the ANN model on the validation set. As indicated in Section 5, this model was used as a submission for task 1. Table 5 shows the confusion matrix of

Table 5: Confusion matrix of the SVM with $n = 4$ and $t = 0.25$ on the validation set

Actual \ Predicted	Positive	Negative	Neutral	Mixed	Total
Positive	838	128	102	173	1,241
Negative	174	1849	292	215	2,530
Neutral	378	633	2106	41	3,158
Mixed	56	39	5	64	164
Total	1,446	2,649	2,505	493	7,093
F-Measure	68.47%				

the best performance of the SVM model for task 2 on the validation set. This configuration was used as a submission for Task 2 (see Section 5).

5 Results and Analysis

For the shared task, we submitted 2 runs for task 1 and 2 runs for task 2.

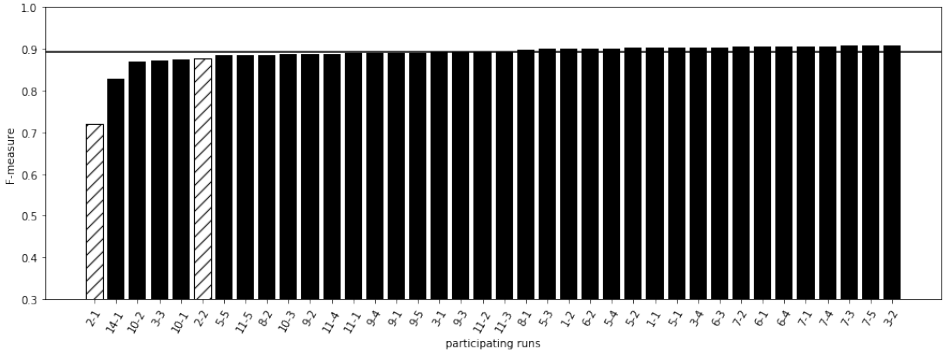
5.1 Runs for Task 1

We submitted 2 runs for task 1: CLaC_T1_run1 and CLaC_T1_run2.

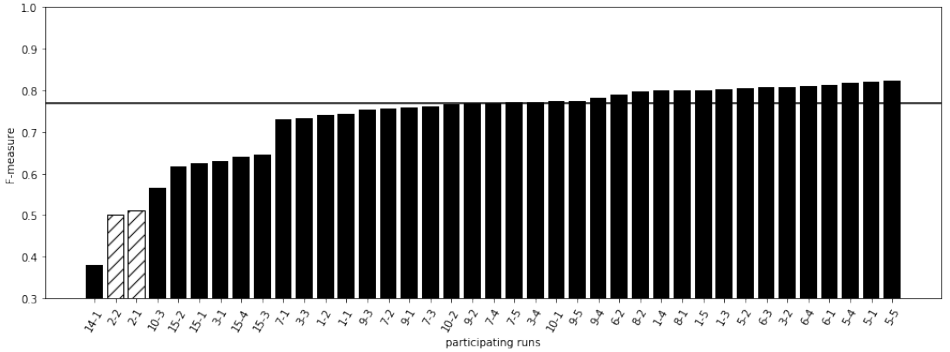
CLaC_T1_run1: Consisted of the best SVM model described in Section 4.2, using the binary bag-of-words approach with word frequency cutoff $n = 4$ and entropy difference cutoff $t = 0.25$. While the performance with the validation set achieved an F-measure of 83.05%, our final result with the test set dropped to 71.90%. Table 6 shows the official results of the SVM run at Task 1.

CLaC_T1_run2: Consisted of the ANN model described in Section 4.2, also using the binary bag-of-words approach with word frequency cutoff $n = 4$ and entropy difference cutoff $t = 0.25$. Although the validation results for the ANN classifier showed an F-measure of 85.54%, our final F-measure with the test set increased to 87.61%. Table 6 shows the official results of the ANN run. Compared to the SVM classifier of CLaC_T1_run1, the ANN achieved better results at classifying *transport* messages, with a precision of 77.96% versus 56.14% for the SVM.

Figure 2 compares the F-measure of all participants for tasks 1 and 2, and indicates the median score by a horizontal line. As the figure shows, for task 1, the SVM (run 2-1 in the Figure) achieved the lowest f-measure among all participants; while the ANN (run 2-2) was close to the median of all participants. This difference in performance during the official runs was surprising, as these these two classifiers achieved very similar results with the validation set. However, as indicated in Table 6, the ANN out-performed the SVM by a difference of more than 16% in F-measure. We suspect that this large gap stems from the fact that, although we fine-tuned the hyper parameters of both models using the validation set, the dropout of the ANN reduced over-fitting. We also believe that the simple feature used in this task, contributed to their low performance.



(a) F-measure for Task-1



(b) F-measure for task-2

Figure 2: F-measure of all participants in task 1 (a) and task 2 (b)

Measure	Run	
	CLaC_T1_run1	CLaC_T1_run2
true positive	4387	6093
false_positive	3428	1723
false_negative	1	0
(micro-mean) precision	0.56136	0.77955
(micro-mean) recall	0.99977	1
(micro-mean) F1-measure	0.719	0.87612

Table 6: Official results for task 1

Measure	Run	
	CLaC_T2_run1	CLaC_T2_run2
true positive	1350	1320
false_positive	2591	2621
false_negative	0	0
(micro-mean) precision	0.34255	0.33494
(micro-mean) recall	1	1
(micro-mean) F1-measure	0.5103	0.50181

Table 7: Official results for task 2

5.2 Runs for Task 2

For task 2, we submitted two runs.

CLaC_T2_run1: Consisted of the SMV using the same binary bag-of-words approach with word frequency cutoff $n = 4$ and entropy difference cutoff $t = 0.25$. The validation results for the classifier showed an F-measure of 68.47%, while the final F-measure dropped to a low 51.03%.

CLaC_T2_run2: Consisted of a similar SVM, but trained on a more powerful machine, allowing us to use $n = 3$. In retrospect, increasing the number of features did not turn out to be a successful approach, as the final F-measure dropped to a low 50.18%, placing us again at the low end of the scores.

Figure 2b, shows that the first SVM (run 2-1) achieved the third lowest F-measure, and our second SVM (run 2-2) achieved the second lowest F-measure. These results were rather surprising as they were approximately 18% lower than those achieved during our validation runs. As with our runs at task 1, we suspect that the models over-fitted the training set due to the large number of features used. With the validation set, the F-measure for mixed tweets was only 19.48% for the SVM classifier, with $n = 4$ and entropy difference $t = 0.25$, much lower than the other three sentiment labels. This seems to show that the SVM classifier for sentiments achieved better results when classifying tweets with a single polarity than those with mixed polarity. As seen in Table 2, the proportion of mixed polarity messages in the training set was significantly lower than the other 3 sentiment labels; this might also have contributed to this low performance.

6 Conclusion

This paper described our first participation to the DEFT shared task. Due to lack of time, we only participated to the first two tasks: transport / non-transport classification and global polarity. We deployed models based on standard hand-crafted features and used off-the-shelf toolkits to pre-process the tweets and experiment with a variety of supervised learning models.

Although our results with the validation set seemed somewhat acceptable, most of our runs underperformed with the actual test set. Our results at the shared tasks clearly indicate that training with the Support Vector Machine classifiers seemed to over-fit the training set with the large feature set that we used; whereas the Artificial Neural Network seemed more robust as is reached 85.54% at task 1 with very little fine-tuning of hyper-parameters.

Acknowledgement

The authors would like to thank the organizers of the DEFT-2018 challenge. We gratefully acknowledge the financial support of NSERC and the NVIDIA Corporation for the donation of the Titan X Pascal GPU used for this research.

References

KOULOUMPIS E., WILSON T. & MOORE J. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of the International AAAI Conference on Web and Social Media, p. 538–541, Barcelona, Spain.

LOPER E. & BIRD S. (2002). NLTK: The Natural Language Toolkit. In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics (ETMTNLP-2002), p. 63–70, Philadelphia, USA.

MOHAMMAD S. M., KIRITCHENKO S. & ZHU X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises (SemEval-2013), p. 321–327, Atlanta, USA.

NAIR V. & HINTON G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-2010), p. 807–814, Haifa, Israel.

PAK A. & PAROUBEK P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the International Conference on Language Resources and Evaluation (LREC-2010), volume 10, p. 1320–1326, Valletta, Malta.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L. & TORRES-MORENO J.-M. (2018). DEFT2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en île de france. In Actes de DEFT, Rennes, France.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M. & DUCHESNAY E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, **12**, 2825–2830.

REITAN J., FARET J., GAMBÄCK B. & BUNGUM L. (2015). Negation scope detection for twitter sentiment analysis. In Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA), a workshop collocated with EMNLP, p. 99–108, Lisboa, Portugal.

SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I. & SALAKHUTDINOV R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, **15**(1), 1929–1958.

TIELEMAN T. & HINTON G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, **4**(2), 26–31.

Modèles en Caractères pour la Détection de Polarité dans les Tweets

Davide Buscaldi[1] Joseph Le Roux[1] Gaël Lejeune[2]
(1) LIPN, Université Paris XIII, 99 avenue JB Clément, 93430, Villetaneuse
(2) STIH, Sorbonne Université, 28 rue Serpente, 75006, Paris
prenom.nom@lipn.univ-paris13.fr, prenom.nom@sorbonne-universite.fr

RÉSUMÉ

Dans cet article, nous présentons notre contribution au Défi Fouille de Textes 2018 au travers de trois méthodes originales pour la classification thématique et la détection de polarité dans des *tweets* en français. Nous y avons ajouté un système de vote. Notre première méthode est fondée sur des lexiques (mots et emojis), les n-grammes de caractères et un classificateur à vaste marge (ou *SVM*). tandis que les deux autres sont des méthodes endogènes fondées sur l'extraction de caractéristiques au grain caractères : un modèle à mémoire à court-terme persistante (ou *BiLSTM* pour *Bidirectionnal Long Short-Term Memory*) et perceptron multi-couche d'une part et un modèle de séquences de caractères fermées fréquentes et classificateur *SVM* d'autre part. Le *BiLSTM* a produit de loin les meilleurs résultats puisqu'il a obtenu la première place sur la tâche 1, classification binaire de *tweets* selon qu'ils traitent ou non des transports, et la troisième place sur la tâche 2, classification de la polarité en 4 classes. Ce résultat est d'autant plus intéressant que la méthode proposée est faiblement paramétrique, totalement endogène et qu'elle n'implique aucun pré-traitement.

ABSTRACT

Character-level Models for Polarity Detection in Tweets

We present our contribution to the DEFT 2018 shared task, with three entries based on different methods to perform topic classification and polarity detection for tweets in French, to which we added a voting system. Our first entry is based on lexicons (for words and emojis), character n-grams and a classifier implemented with a support vector machine (*SVM*), while the other two are endogenous methods based on character-level feature extraction : first a long short-memory recurrent neural network (*BiLSTM*) feeding a classifier implementing a multi-layer perceptron, and second a model based on frequent closed character sequences with a SVM. The *BiLSTM* system gave the best results by far. It ranked first on task 1, a binary theme classification task, and third on task 2, a four-class polarity classification task. This result is very encouraging as this method has very few priors, is completely endogenous, and does not require any specific preprocessing.

MOTS-CLÉS : Analyse en Caractères, Bi-LSTM, n-grammes de caractères, Détection de Polarité, Analyse de Tweets.

KEYWORDS: Character-level Models, Bi-LSTM, character n-grams, Polarity detection, Tweet Analysis.

1 Introduction

L'analyse automatique des *tweets* est un domaine très actif du Traitement Automatique des Langues (TAL), notamment sur l'aspect étude de la polarité et détection des émotions. Ceci n'est pas très étonnant dans la mesure où les *tweets*, et plus généralement les microblogs, constituent un moyen de jauger l'opinion individuelle d'une grande quantité de personnes. Il existe donc un grand nombre d'applications qui pourraient profiter de systèmes performants pour cette tâche. D'un autre côté, la classification binaire de *tweets* selon qu'ils relèvent ou non d'une thématique particulière est une tâche connexe à la détection des émotions mais souvent délaissée, et placée en amont de la chaîne de traitement de TAL. Elle est traditionnellement implémentée par filtrage sur des chaînes de caractères ou des expressions rationnelles qui font référence à des termes du thème visé. C'est regrettable car connaître la polarité d'une émotion est intéressant mais ça l'est plus encore si l'on sait à quoi le *tweet* fait référence.

D'un côté la tâche 1 consiste donc à décider si un *tweet* traite ou non des transports et de l'autre côté la tâche 2 pour laquelle, étant donné un *tweet* traitant des transports, il faut détecter la polarité : Neutre, Négatif, Positif ou Mixte. L'article décrivant cette édition du Défi Fouille de Textes propose une description précise du processus de collecte et d'annotation du corpus (Paroubek *et al.*, 2018), nous ne reviendrons donc pas plus ici sur les données d'entrée.

Dans la section 2 nous décrirons les trois méthodes que nous avons mises en place ainsi qu'un système de vote entre les méthodes. Ensuite, nous présenterons dans la section 3 les résultats que nous avons obtenus dans la compétition officielle ainsi que des expériences plus complètes sur nos méthodes. Enfin, nous proposerons quelques conclusions et perspectives sur ce travail dans la section 4.

2 Méthodes : trois modèles en caractères et un système de vote

2.1 Méthode 1 : Lexiques et caractères

Le premier système est basé sur des SVM avec un noyau gaussien, où les vecteurs comportent un mélange de caractéristiques à niveau de caractères et lexicales, à l'aide de dictionnaires.

Pour les caractères, nous avons choisi d'utiliser, en tant que caractéristiques, tous les n-grammes de caractères (espaces exclues) de taille comprise entre 3 et 6, avec une fréquence minimale de 100 (arbitrairement choisie) dans le corpus d'entraînement. Ce choix est fondé sur l'observation que les n-grammes de caractères permettent de capturer des variations sur un même mot ou acronyme qui sont fréquentes dans le langage des *tweets*, par exemple : *#ratp*, *@ligne7_ratp*, *#merciratp*, *@grouperatp*, etc. Le poids d'un n-gramme est de 0 s'il n'apparaît pas dans le *tweet* ou $s(n) = \sum_{i=1}^{nbOcc} 1 + pos(n_i)/len(t)$ s'il apparaît dans le *tweet*, où $pos(n_i)$ indique la position du premier caractère de l'i-ème occurrence du n-gramme n et $len(t)$ est la taille du *tweet* en nombre de caractères.

Nous avons choisi de garder aussi dans le modèle une composante lexicale, avec l'utilisation de dictionnaires de polarité, en particulier les dictionnaires labMT (Dodds *et al.*, 2011) et FEEL (Abdaoui *et al.*, 2017). Les dictionnaires ont été utilisés pour associer deux poids à chaque *tweet* : un poids pour la première moitié du *tweet* et un autre poids pour la deuxième moitié. Les scores des dictionnaires ont été normalisés dans l'intervalle $[-1, +1]$, dans le cas du FEEL cela revient à associer le label *négatif* à -1 et *positif* à $+1$. Pour labMT, étant donné que les scores sont dans l'intervalle $(1, 9)$, on

a associé le minimum à -1 et le maximum à $+1$ puis on a mis à l'échelle les valeurs intermédiaires. Finalement pour calculer la valeur des caractéristiques pour chaque moitié du *tweet* on garde la somme des scores des mots.

Pour la tâche T2, le système a utilisé les résultats de la tâche T1 pour travailler uniquement sur les *tweets* étiquetés *Tranport*. On a donc créé un modèle de polarité pour détecter si les *tweets* avaient une polarité ou pas. La sortie de ce modèle a été utilisé comme entrée pour un modèle dédié à la polarité positive et un modèle dédié à la polarité négative. Finalement nous avons produit un script pour combiner les résultats produits par chaque modèle. Donc si un *tweet* avait été reconnu comme *Transport*, alors on vérifie s'il est polarisé ou pas. S'il est polarisé, on vérifie s'il est *positif* ou pas, on vérifie s'il est *négatif* ou pas, et s'il est *positif et négatif* en même temps on lui assigne l'étiquette *MIXPOSNEG*.

2.2 Méthode 2 : BiLSTM

Notre deuxième système utilise des réseaux de neurones récurrents pour implanter un classificateur, plus spécifiquement les LSTM (Hochreiter & Schmidhuber, 1997) qui sont largement utilisés en traitement automatiques des langues. La classification se fait en trois temps :

1. Le texte est séparé aux espaces. Chaque segment est traité comme une séquence d'octets lue de gauche à droite et de droite à gauche par deux réseaux récurrents *niveau caractère*. Les vecteurs résultats des lectures sont additionnés et servent de représentation du segment, dite compositionnelle. Pour une séquence de caractères $s = c_1 \dots c_m$, on calcule pour chaque position $h_i = LSTM_o(h_{i-1}, e(c_i))$ et $h'_i = LSTM_{o'}(h'_{i+1}, e(c_i))$, où e est la fonction de plongement des caractères vers les vecteurs denses, et $LSTM$ est un raccourci pour une fonction implantant la cellule récurrente des LSTM. La représentation compositonnelle du segment est $c(s) = h_m + h'_1$
2. La séquence de segments est lue à nouveau de gauche à droite et de droite à gauche par de nouveaux réseaux récurrents *niveau mot* qui prennent en entrée pour chaque segment la représentation compositionnelle venant de l'étape précédente à laquelle on ajoute une représentation vectorielle du segment si celui-ci était présent plus de 10 fois dans le corpus d'entraînement. Pour une séquence de segments $p = s_1 \dots s_n$, on calcule $l_i = LSTM_m(l_{i-1}, c(s_i) + e(s_i))$, $l'_i = LSTM_{m'}(l_{i+1}, c(s_i) + e(s_i))$, où c est la représentation compositionnelle donnée ci-dessus et e la fonction de plongement que l'on étend aux segments vus dans l'ensemble d'entraînement. Les états finaux obtenues après lecture dans les deux directions sont sommés et servent de représentation de la phrase d'entrée, $r(p) = l_n + l'_1$.
3. La représentation obtenue sert d'entrée à un perceptron multi-niveau qui effectue la classification finale, aussi bien pour la tâche 1 que pour la tâche 2 : $o(p) = \sigma(O \times \max(0, (W \times r(p) + b)))$ où σ est l'opérateur *softmax*, W, O des matrices et b un vecteur. On interprète la sortie comme une distribution de probabilité sur les classes de *tweets*.

Cette interprétation probabiliste nous permet de réduire l'apprentissage des paramètres du système (*ie.* les plongements de caractères et de segments fréquents, O, W, b, ainsi que les paramètres des 4 cellules LSTM) à la maximisation de la vraisemblance du corpus d'entraînement. On utilise l'algorithme AMSgrad (Reddi *et al.*, 2018) pour calculer la taille du pas lors de la descente de gradient. Pour éviter le sur-apprentissage, nous procédons aux deux ajustements suivants.

— On écarte aléatoirement du corpus d'entraînement 10% des phrases qui sont utilisées comme ensemble de validation, ce qui permet de décider quand les paramètres sont toujours utiles sur

des données inconnues.
— on utilise la technique du *dropout* (Srivastava *et al.*, 2014), sur tous les vecteurs à chaque étage du réseau — sauf la couche finale bien sûr. Durant l'apprentissage les neurones sont aléatoirement remis à zéro avec une probabilité de 0,5.

Les plongements de caractères sont de taille 16, ceux des segments de taille 32, la couche d'entrée du perceptron est de taille 64 et la couche cachée de taille 32. La couche de sortie est de taille 2 pour la tâche 1, et 4 pour la tâche 2. Nous utilisons la bibliothèque DYNET [1] avec les paramètres par défaut.

2.3 Méthode 3 : motifs en caractères

Notre troisième système utilise également une approche d'analyse au grain caractère. Plus exactement cette approche se situe à la lisière entre l'algorithmique du texte et la fouille de données. Nous utilisons des motifs (en caractères) fermés et fréquents comme traits pour entraîner un classificateur.

Les propriétés de fermeture et de fréquence sont définis de la façon suivante :

Fermeture : le motif ne peut être étendu vers la gauche ou vers la droite sans diminuer son nombre d'occurrences

Fréquence : le motif respecte une borne minimale de nombre d'apparitions

Pour calculer des motifs en caractères de manière efficace, en l'occurrence avec une complexité linéaire en la taille des données, nous utilisons ici une implantation en Python de l'algorithme décrit dans (Ukkonen, 2009) exploitant les tableaux de suffixes augmentés décrits dans (Kärkkäinen *et al.*, 2006). (Buscaldi *et al.*, 2017) rappellent que les propriétés de fermeture et de fréquence correspondent en algorithmique du texte aux propriétés de maximalité et de répétition. Du point de vue du TAL, cette technique peut être décrite comme une segmentation *non-supervisée* en ce sens que les règles de découpage ne sont pas pré-définies mais sont calculées en fonction du corpus donné en entrée.

De façon traditionnelle en fouille de données, un défi important est de limiter l'explosion du nombre de motifs, que ce soit pour des raisons calculatoires ou pour des raisons de lisibilité des résultats.Un filtrage classique consiste à appliquer aux motifs deux types de contrainte :

— La contrainte de support par laquelle on définit le nombre minimal ($minsup$) et maximal ($maxsup$) d'objets qui supportent un motif. Ici cela consiste à définir le nombre minimal et maximal de *tweets* dans lequel le motif apparaît.
— La contrainte de longueur par laquelle on définit la longueur minimale ($minlen$) et maximale ($maxlen$) d'un motif. Ici, il s'agit d'une longueur en caractères.

Notre chaîne de traitement est définie comme suit :

1. Calcul des motifs fermés fréquents dans tout le corpus de *tweets* (train+test) ;
2. Filtrage des motifs selon la longueur ;
3. Filtrage des motifs selon le support ;
4. Représentation de chaque *tweet* sous forme d'un vecteur d'effectif des motifs ;
5. Utilisation de l'implantation de SCIKIT du SVM ONE VS REST.

Durant la phase d'entraînement nous avons testé la méthode au travers d'un validation croisée en 10 strates au moyen de la fonction STRATIFIEDKFOLD de SCIKIT [2]. En examinant les résultats, nous avons observé que la qualité des résultats n'augmentait plus au delà d'un seuil de $maxlen$ de

1. https://github.com/clab/dynet
2. http://scikit-learn.org/

5. Nous avons également remarqué, ce qui est conforme à des résultats précédents sur des données comparables (Buscaldi *et al.*, 2017) que l'application de contraintes de support avait assez peu d'influence sur les résultats. Nous avons donc choisi de ne pas chercher à paramétrer finement cette contrainte. Toutefois, il est à noter que la réduction de l'espace de description engendrée par cette contrainte permet de diminuer le coût en calcul.

Les résultats que nous avons soumis pour la phase de test ont été obtenus avec les configurations suivantes :

— Pas de taille minimale ($minlen = 1$) ;
— Pour la tâche 1 $maxlen = 2$ et pour la tâche 2 $maxlen = 3$;
— Pas de contrainte de support ;
— Utilisation d'un noyau linéaire.

La configuration ci-dessus s'est avérée la plus efficace sur les données d'entraînement et la plus fiable pour effectuer les calculs dans le temps imparti. L'utilisation d'un noyau radial permettait toutefois d'obtenir de bons résultats y compris en se contentant des motifs de taille 1 ($maxlen = 1$).

2.4 Systèmes de vote

Avec notre quatrième run, nous avons tenté de tirer profit des propriétés respectives de nos trois méthodes en élaborant un système de vote. Pour les deux tâches, il s'agit simplement d'un vote majoritaire entre les résultats des trois méthodes. Pour la tâche 2 qui comporte 4 classes, il a fallu gérer les cas d'égalité qui représentaient près de 25% des cas. Nous avons considéré que l'absence d'accord tangible entre les méthodes indiquait que nous avions à faire à des *tweets* sans tendance particulière. Nous avons donc donné l'étiquette *MIXPOSNEG*. Les cas d'indécision étant plutôt fréquents, cela a abouti à une sur-représentation de cette classe qui a nui aux résultats comme nous le montrerons dans la section suivante.

3 Résultats

3.1 Tâche 1 : classification thématique *Transport* ou *Inconnu*

Cette tâche était relativement facile avec beaucoup de configurations au-delà des 80% de F-mesure. Nos systèmes se sont bien comportés et le BiLSTM a même obtenu la première place. Les résultats officiels figurent dans le tableau 1.

Run	Précision	Rappel	F-mesure	VP	FP	FN
Run1 (Lexique et caractères)	0,80463	1.0	0,89174	6289	1527	0
Run2 (BiLSTM)	0,831246497	1.0	0,90785	6497	1319	0
Run3 (motifs en caractères)	0,77364	0,999	0,87231	6046	1769	1
Run4 (vote)	0,824826446	0,999	0,90394	6446	1369	1

TABLE 1 – Résultats officiels de Tweetaneuse sur la tâche 1

Assez logiquement au vu de la facilité de la tâche, nos trois systèmes ont souvent été en accord complet (71,9% des cas précisément). Le système de vote n'a pas fonctionné aussi bien que nous

l'aurions souhaité car nous n'avons pas pu détecter de réelle complémentarité entre les systèmes. Les systèmes 1 et 3 ont rarement eu raison *contre* le BiLSTM (40 cas soit seulement 0,5% des cas). Dans 320 cas (soit 4,1%), il y a eu uniquement un de nos systèmes qui a été correct. Enfin, dans 1049 cas (soit 13,41%), aucun de nos systèmes n'avait trouvé la bonne étiquette, dont 93 reprises pour la catégorie transports, ce qui est assez attendu puisque la classe *Inconnu* s'était avérée la plus difficile à détecter dans la phase d'entraînement.

Derrière ces erreurs nous pouvons les difficultés inhérentes à un tâche d'annotation même binaire. Toutefois, nous avons pu voir des *tweets* dont la classification était vraiment étonnante.

Quelques exemples étiquetés dans la catégorie *Transport* mais classés *Inconnu* par nos 3 systèmes :

- Remember à la #CDM2014 quand un tracteur est passé dans toutes les rues de Berche pour nous récupérer et chanter la Marseillaise
- Malheureusement pour la #fra ce soir les Marines US ne débarqueront pas par la mer pour la libérer contre la #ger ...#EURO2016
- @scanlan75018 depuis qu' il a eu le coup il n' arrive plus à accélérer mais comme c' est une finale il à essayer de forcer

Quelques exemples étiquetés dans la catégorie *Inconnu* mais classés *Transport* par nos 3 systèmes :

- La prochaine fois que la SNCF me met dans le sens inverse de la marche alors qu' il reste plein de places dans le train je vomis partout.
- @LIGNEJ_SNCF #ligne Merci au conducteur qui vient de partir de houilles de m'avoir attendu
- Tu pars bcp plus tard le matin et bcp plus tôt le soir. Ben ça change rien c' est l' horreur sur la @Ligne1_RATP Super mois d' août à la #ratp
- #duflot2017 veut rouler en bus au poireau bio? enfin la politique me fait rire!! Celui de son mec ne l'est pas?? MDR
- En même temps , y' a qu' un bus pour toute la journée , il n' allait quand même pas passer

En analysant brièvement les *tweets* pour lesquels aucun de nos systèmes n'a trouvé la bonne classe, nous avons pu voir quelques cas intéressants. Les *tweets* mêlant bus et football contenaient souvent une référence que nos systèmes n'avaient pas pu modéliser : *mettre le bus* dans le sens de se ruer en défense et *rester dans le bus* en référence au bus de Knysna lors de la Coupe du Monde 2010. Un autre cas est le *tweet* ou il n'est pas réellement question de transports mais où le transport sert au mieux de toile de fond. Il s'agit par exemple de cas où le twittos indique explicitement qu'il se situe dans les transports sans que cela ait aucun lien avec ce dont il est train de parler. En voici quelques exemples :

- c pas je marchais sereine pour aller prendre mon bus et je remarque y' a un bouton de ma chemise ouvert , dévoilant tt mon soutif
- à l' arrêt de bus g vu mon prof de philo j' étais pas sereine
- Jsuis encore à l' arrêt de bus et je pense à ce qui m' attend ce soir niveau révisions ptdr go mourir ===>
- Hier dans le métro j' ai vu un mec il avait des mains!! s' il te donne une gifle t' es mal...

3.2 Tâche 2 : classification en termes de polarité

Dans le tableau 2 sont recensés nos résultats avec les métriques officielles. Une fois encore, le modèle BiLSTM s'est avéré le plus compétitif avec une troisième place sur cette tâche. Dans cette tâche, le différentiel entre nos système s'est creusé et nous avions un triple accord dans seulement 1033 cas (soit 26,2% des cas). Dans 1391 cas (35,3%) deux des systèmes ont fourni la bonne réponse, influent positivement les résultats du système de vote. Enfin, pour 622 *tweets* (15,78%) aucun de nos systèmes n'a trouvé la bonne classe.

Run	Précision	Rappel	F-mesure	VP	FP	FN
Run1 (Lexique et caractères)	0,47599	0,93313	0,63041	1814	1997	130
Run2 (BiLSTM)	0,67699	1	0,80738	2668	1273	0
Run3 (Motifs en caractères)	0,57828	1	0,7328	2279	1662	0
Run4 (vote)	0,62766	0,9996	0,77113	2473	1467	1

TABLE 2 – Résultats officiels de Tweetaneuse sur la tâche 2

Nous présentons dans le tableau 3 les résultats par classe pour chacune de nos méthodes. Nous indiquons également la macro F-mesure qui permet de mettre un peu en lumière les configurations qui s'accommodent le mieux de la classe minoritaire *MIXPOSNEG*. Cette classe s'est révélée très difficile à prédire en plus de son faible effectif du fait de sa définition forcément plus imprécise. Nous pouvons observer que le BiLSTM a fait la différence sur toutes les classes. Il s'est aussi mieux comporté que les autres dans la détection des *tweets* de la classe *MIXPOSNEG*.

Parmi les erreurs que nos systèmes ont commis, nous avons extrait quelques exemples pour lesquelles la prédiction n'était pas évidente.

Voici une première série de *tweets* pour lesquels nos 3 systèmes ont prédit *NEGATIF* mais où l'annotation était *MIXPOSNEG* :

- `RT @allenlafrance : #PLC veulent se débarrasser des Postes,on va leur suggérer faire de même avec RC,CBC. et gérer les affaires du Pays. htt...`
- `- Le contrôleur lui dit que ça doit être un pb de date et le mec sûr de lui sort ' Ah c' est pas de ma faute ' , il sort donc son billet et dit-`
- `La #SNCF a vraiment un des pire service clients que je connaisse ! C est boite est une honte vivement la concurrence`

Quelques exemples de *tweets* pour lesquels nos 3 systèmes ont prédit *POSITIF* mais où l'annotation était *MIXPOSNEG* :

- `jsuis morte laura elle est descendu de son bus juste pour me dire bonjour à arrêt et remonter mdrrrrr`
- `@J_Lutecia @Virginiement Yep alors par contre j' ai cette contrainte de dernier bus qui fait que ça serait cool si on pouvait faire ça tot :3`
- `les gens ils pensent jsuis l' genre à courir après pour les capter PTDRRRR archi drôle , déjà que j' cours pas après mon bus alors vous nvm`

CLASSE	Précision	Rappel	F-mesure	VP	FP	FN
run1 (Micro F1 : 0,4761, Macro F1 : 0,3717)						
NEUTRE	**0,7783**	0,1414	0,2393	179	51	1087
POSITIF	0,5856	0,4145	0,4855	342	242	483
NEGATIF	0,5136	**0,8393**	0,6373	1243	1177	238
MIXPOSNEG	0,0882	0,2125	0,1247	51	527	189
run2 (Micro F1 : 0,677, Macro F1 : 0,5946)						
NEUTRE	0,6854	**0,75**	**0,7162**	978	449	326
POSITIF	**0,6254**	**0,6429**	**0,6341**	551	330	306
NEGATIF	**0,7306**	0,7043	0,7172	1074	396	451
MIXPOSNEG	**0,3988**	0,2549	**0,311**	65	98	190
run3 (Micro F1 : 0,5783, Macro F1 : 0,4872)						
NEUTRE	0,5741	0,7132	0,6361	930	690	374
POSITIF	0,5591	0,4527	0,5003	388	306	469
NEGATIF	0,6557	0,6007	0,627	916	481	609
MIXPOSNEG	0,1957	0,1765	0,1856	45	185	210
run4 (Micro F1 : 0,6278, Macro F1 : 0,5475)						
NEUTRE	0,7474	0,6127	0,6734	799	270	505
POSITIF	0,6973	0,5134	0,5914	440	191	417
NEGATIF	0,6957	0,7541	**0,7237**	1150	503	375
MIXPOSNEG	0,1446	**0,3333**	0,2017	85	503	170

TABLE 3 – Résultats détaillés classe par classe pour chaque run de la tâche 2

4 Discussion

Dans cet article, nous avons présenté trois méthodes exploitant le grain caractère pour la classification de *tweets* et la détection de polarité. Ces trois méthodes utilisent de l'apprentissage supervisé. La première méthode combine ressources exogènes, des lexiques, et n-grammes de caractères, la seconde exploite un BiLSTM tandis que la troisième utilise des motifs en caractères fermés fréquents.

Les bons résultats obtenus, en particulier pour le BiLSTM montrent que l'utilisation du grain caractère pour les tâches de TAL est amenée à avoir un impact important. En effet, les méthodes en caractère ont l'avantage de bien se comporter dans des contextes bruités, ce qui est le cas ici puisque les variations de graphie et la distanciation avec les normes syntaxiques sont des caractéristiques majeures des *tweets*. Par ailleurs, ces approches permettent de se passer de pré-traitement ce qui permet de simplifier les chaînes de traitement et par conséquent de réduire les erreurs en cascade(Lejeune *et al.*, 2015). Enfin, ces méthodes permettent de capturer de manière purement endogène des informations sur la structure intra-token (morphèmes) comme sur la structure extra-token (expressions multi-mots par exemple) ce qui peut être particulièrement intéressant dans un contexte impliquant plusieurs langues ou des langues peu dotées en ressources. Nos prototypes sont disponibles en ligne [3] et librement utilisables.

3. https://github.com/rcln/tweetaneuse2018

Références

ABDAOUI A., AZÉ J., BRINGAY S. & PONCELET P. (2017). FEEL : a French Expanded Emotion Lexicon. *Language Resources and Evaluation*, **51**(3), 833–855.

BUSCALDI D., GREZKA A. & LEJEUNE G. (2017). Tweetaneuse : Fouille de motifs en caractères et plongement lexical à l'assaut du deft 2017. In *Actes du 13e Défi Fouille de Texte*, p. 65–76, Orléans, France : Association pour le Traitement Automatique des Langues.

DODDS P. S., HARRIS K. D., KLOUMANN I. M., BLISS C. A. & DANFORTH C. M. (2011). Temporal patterns of happiness and information in a global social network : Hedonometrics and twitter. *PloS one*, **6**(12), e26752.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

KÄRKKÄINEN J., SANDERS P. & BURKHARDT S. (2006). Linear work suffix array construction. *Journal of the ACM*, p. 918–936.

LEJEUNE G., BRIXTEL R., DOUCET A. & LUCAS N. (2015). Multilingual event extraction for epidemic detection. *Artificial Intelligence in Medicine*. doi : 10.1016/j.artmed.2015.06.005.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L. & TORRES-MORENO J.-M. (2018). Deft2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en île de france. In *Actes de DEFT*, Rennes, France.

REDDI S. J., KALE S. & KUMAR S. (2018). On the convergence of adam and beyond. In *International Conference on Learning Representations*.

SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I. & SALAKHUTDINOV R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**(1), 1929–1958.

UKKONEN E. (2009). Maximal and minimal representations of gapped and non-gapped motifs of a string. *Theoretical Computer Science*, p. 4341–4349.

Concaténation de réseaux de neurones pour la classification de tweets, DEFT2018

Damien Sileo[1, 2,*] Tim Van de Cruys[2,*] Philippe Muller [2] Camille Pradel [1]
(1) Synapse Développement, 5 Rue du Moulin Bayard, 31000 Toulouse
(2) IRIT, Université Paul Sabatier 118 Route de Narbonne 31062 Toulouse
(*) Contributions égales
damien.sileo@synapse-fr.com, camille.pradel@synapse-fr.com,
philippe.muller@irit.fr, tim.van-de-cruys@irit.fr

RÉSUMÉ
Nous présentons le système utilisé par l'équipe Melodi/Synapse Développement dans la compétition DEFT2018 portant sur la classification de thématique ou de sentiments de tweets en français. On propose un système unique pour les deux approches qui combine concaténativement deux méthodes d'embedding et trois modèles de représentation séquence. Le système se classe 1/13 en analyse de sentiments et 4/13 en classification thématique.

ABSTRACT
Concatenation of neural networks for tweets classification, DEFT2018

We present the system used by the Melodi / Synapse Development team in the DEFT2018 competition on the classification of themes or sentiments of tweets in French. We propose a unique system for both approaches that combines concatentively two embedding methods and three sequence representation models. The system ranks 1/13 in sentiment analysis and 4/13 in thematic classification.

MOTS-CLÉS : *Fasttext*, classification, ensemble.

KEYWORDS: *Fasttext*, classification, ensemble.

1 Introduction

La classification de textes est une application importante du traitement des langues, instanciée sur de nombreux aspects : identification de langue, analyse de sentiment, détection de contenu haineux, catégorisation thématique pour n'en citer que quelques uns. Son application à la communication sur les réseaux sociaux tels que Twitter a néanmoins nécessité quelques déclinaisons permettant d'appréhender au mieux la concision et la "liberté" du langage. La constante évolution des techniques de classification complique leur comparaison objective, en particulier sur des domaines différents comme le français.

La campagne DEFT2018 (Paroubek *et al.*, 2018) propose 4 taches liées au TALN sur des tweets en français, dont 2 taches de classification où chaque tweet est associé à une classe y.

1. La classification thématique des tweets : $y \in \{\text{TRANSPORT}, \text{INCONNU}\}$.
2. L'analyse de sentiments des tweets : $y \in \{\text{POSITIF}, \text{NEGATIF}, \text{MIXPOSNEG}, \text{NEUTRE}\}$. MIXPOSNEG concerne les sentiments partagés et NEUTRE les sentiments peu marqués.

On propose d'évaluer des systèmes généraux ayant la même configuration d'hyperparamètres sur les deux taches.

2 Modèle

2.1 Vue d'ensemble

On utilise un système avec deux niveaux de représentations. D'abord les mots sont représentés par deux composantes :

— Des embeddings fixés (représentations issues du modèle *Fasttext*, qui prennent en compte la morphologie des mots) ;
— Des embeddings appris.

Cette approche dite multi-canaux (Kim, 2014) permet d'exploiter à la fois les régularités du corpus d'entraînement non supervisé et des données DEFT dans les représentations de mots.

Puis ces composantes partagent le rôle d'entrée pour trois modèles de représentation de séquences :

— Un réseau de convolution 1D profond, dont l'architecture sera détaillée. Ce modèle permet de détecter des motifs pertinents pour la classification.
— La moyenne de tous les embeddings de mots présents dans la phrase, suivie d'une projection et d'une non-linéarité. Cette composante permet de prendre en compte également tous les mots de la phrase et de représenter un aspect thématique/contextuel plus global.
— Un réseau récurrent (GRU (Chung *et al.*, 2014)) qui est le modèle le plus général des trois, capable de capturer d'autres statistiques de la nature séquentielle des tweets.

Ce système est entraîné 15 fois pour chaque tache avec des initialisations différentes sur 90% des données d'entraînement. Les 8 meilleurs systèmes d'après le score F1 sur les données de validation restantes sont retenues. La prédiction est alors réalisée avec la moyenne des estimations de probabilité par classe fournies par chaque modèle. Par ailleurs la partie fixe des embeddings est issue d'un apprentissage de *Fasttext* à chaque fois différente. Cette stratégie sert à augmenter la variance entre les modèles pour améliorer l'ensemble.

La figure 1 montre l'architecture utilisée. Les nombres entre crochets sont les dimensions.

2.2 *Fasttext*

Le modèle skipgram *Fasttext*(Schmidhuber, 2015) est basé sur le modèle skipgram de *word2vec* (Mikolov *et al.*, 2013), qui consiste à apprendre des représentations de mots pour qu'elles optimisent une tâche de prédiction du contexte des mots. La différence principale est que la représentation h_w d'un mot w ne se résume plus à u_w, la représentation de son symbole. Elle est augmentée de la représentation des n-grammes de caractères contenus dans w, nommés $u_g, g \in \mathcal{G}_w$:

$$h_w = u_w + \sum_{g \in \mathcal{G}_w} u_g \tag{1}$$

$\mathcal{G}_w$ correspond aux n-grammes de w suffisamment fréquents et d'une taille adéquate. La morphologie de w est donc partiellement prise en compte dans h_w, même si l'ordre des grammes est ignoré.

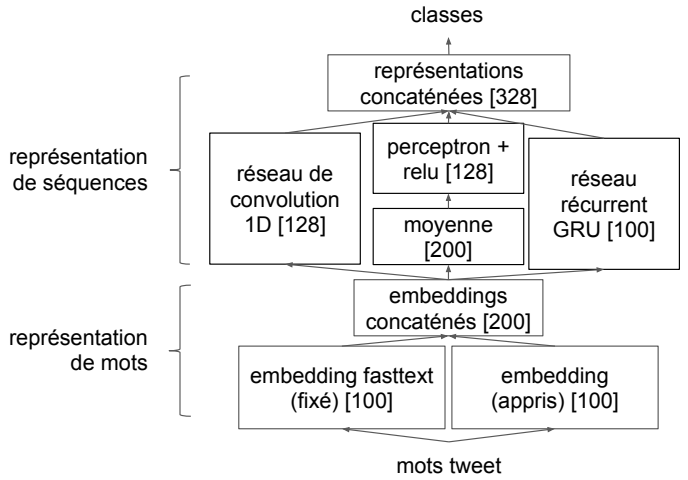


FIGURE 1 – Architecture du système proposé

2.3 Réseau de convolution

Pour l'architecture convolutive, nous nous sommes appuyés sur le modèle décrit par Kim (2014). Dans une couche convolutionnelle, un certain nombre de *feature maps* (filtres) sont appliquées à une fenêtre de mots c (représentés comme des embeddings). Nous appliquons ensuite un *max-over-time pooling* au résultat de la convolution, qui sélectionne la valeur maximale dans une fenêtre particulière en tant que caractéristique correspondant à un filtre particulier.

Il a été démontré que les architectures convolutionnelles profondes améliorent la classification des textes (Conneau *et al.*, 2017) ; nous utilisons un total de trois couches convolutives (128 filtres avec une taille de fenêtre c de 2), suivies de trois étapes de pooling. Dans les deux premières étapes de pooling, nous regroupons deux valeurs ; la dernière étape de pooling est un pooling global sur l'ensemble du contexte, de sorte que nous nous retrouvons avec une valeur unique pour chacun des 128 filtres. La représentation résultante est envoyée à une couche dense (également à 128 valeurs), et une couche finale softmax est utilisée pour la classification.

L'architecture spécifique du modèle et les hyperparamètres ont été choisis en fonction de la performance sur un ensemble de validation.

3 Expériences

3.1 Pré-entraînement des embeddings

Pour apprendre les représentations de mots de *Fasttext*, nous avons utilisé des tweets stockés sur la plateforme *OSIRIM* [1] de l'IRIT qui collecte 1% du flux de Twitter depuis Septembre 2015. À cela s'ajoutent les données de DEFT2018 issues des phases de train et de test. Les tweets sont dé-

1. http://osirim.irit.fr/site/fr/articles/corpus

paramètres	**valeur**
learning rate	0.02
dimensions	100
context window size	5
epochs	4
min_count	5
negative/positive samples ratio	5
loss	negative sampling
minimum n-gram size	3
maximum n-gram size	6
sampling threshold	10^{-4}

TABLE 1 – Paramètres du modèle *Fasttext*

dupliqués, passés en minuscules, et les liens ou occurrences des *[ASCII012CTRLC]* sont remplacés par des symboles unicodes rares (Å et $\mathbb{U}$) pour que leurs caractères ne polluent pas celles des représentations de *Fasttext*. L'espace précédents les apostrophes dans les données de DEFT2018 est enlevé. L'ensemble résultant totalise $50M$ tweets. Les paramètres utilisés par *Fasttext* sont résumés dans la table 1.

3.2 Méthodologie et autres hyperparamètres

La taille du vocabulaire est fixée aux $50k$ mots les plus fréquents dans les données de DEFT2018 (train et test). Seuls les espaces sont utilisés pour réaliser la tokenization.

Le seul hyperparamètre choisi avec validation croisée rigoureuse est la régularisation L2 du softmax final choisie dans $\{10^p, p \in [[-12, -4]]\}$. 10^{-12} a été retenu.

Un dropout de 0.3 est utilisé après les embeddings, ainsi qu'après les 3 systèmes de représentation de séquences.

Les paramètres sont appris par deux algorithmes d'optimisation utilisés successivement : 2 époques avec Adam (Kingma *et al.*, 2014), avec les paramètres par défaut, puis 1 époque de descente de gradient classique avec un taux d'apprentissage de 10^{-5}. La norme des gradients est seuillée de sorte à ne pas dépasser 3.

4 Evaluation

Le tableau 4 présente les résultats fournis par le système d'évaluation pour nos différents runs, puis des statistiques sur les meilleurs systèmes de chaque équipe à titre de comparaison. L'entraînement joint consiste à entraîner les modèles sur T1 et T2 en même temps, et en inférence à considérer seulement les catégories de la tache considérée. Le GRU a été enlevé dans les deux premiers runs. Le résultat de cette ablation est intéressante puisque les opérations du GRU n'étant pas parallélisables, il ralentit significativement les calculs. Pourtant, il n'améliore pas les résultats sur la tache 1 et seulement ponctuellement sur la tache 2. L'apport de l'entraînement joint n'est stable ni en changeant

	T1	T2
1- Entrainements séparés sans GRU	**0.90371**	0.82165
2- Entrainement joint sans GRU	0.90232	0.80413
3- Entrainements séparés	0.90155	0.81918
4- Entrainement joint	0.88367	**0.82288**
médiane des concurrents	0.895025	0.77304
meilleur ou meilleur suivant	**0.90739**	0.81313

TABLE 2 – Résultats des runs. La médiane est celle des meilleurs runs de chaque équipe, et on reporte également le score de la meilleure équipe pour la tache T1 et deuxième pour la tache T2.

la tache ni en enlevant/ajoutant le GRU.

Parmi les systèmes résultants, 3 auraient gagné la compétition restreint à la tache 2, dont le système 1, également classé 4/13 sur la tache 1 et qui semble être le plus robuste.

5 Conclusion

Nous avons décrit un système général présenté à la compétition DEFT2018. En restant général, il serait intéressant d'évaluer l'apport de méthodes d'apprentissage non supervisées traitant l'ordre des mots (Kiros *et al.*, 2015; Nie *et al.*, 2017). Une optimisation plus rigoureuse et exhaustive des hyperparamètres, ou des techniques de maximisation du score F1 (Chase Lipton *et al.*, 2014) pourraient également améliorer les résultats. Enfin, la détection de tweets liés au transport pourrait très sûrement bénéficier de représentations adaptées de mots rares, spécifiques ou de sigles, même si Fasttext traite partiellement ce problème.

Références

CHASE LIPTON Z., ELKAN C. & NARAYANASWAMY B. (2014). Thresholding Classifiers to Maximize F1 Score. *ArXiv e-prints*.

CHUNG J., GULCEHRE C., CHO K. & BENGIO Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv*, p. 1–9.

CONNEAU A., SCHWENK H., BARRAULT L. & LECUN Y. (2017). Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, p. 1107–1116, Valencia, Spain : Association for Computational Linguistics.

KIM Y. (2014). Convolutional Neural Networks for Sentence Classification.

KINGMA D., REZENDE D. & WELLING M. (2014). Semi-supervised Learning with Deep Generative Models. In *arXiv preprint arXiv : ...*, p. 1–9 : Nips.

KIROS R., ZHU Y., SALAKHUTDINOV R. R., ZEMEL R., URTASUN R., TORRALBA A. & FIDLER S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, p. 3294–3302.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Nips*, p. 1–9.

NIE A., BENNETT E. D. & GOODMAN N. D. (2017). DisSent : Sentence Representation Learning from Explicit Discourse Relations.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L. & TORRES-MORENO J.-M. (2018). Deft2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en île de france. In *Actes de DEFT*, Rennes, France.

SCHMIDHUBER J. (2015). On Learning to Think : Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models. *arXiv :1604.00289v1[cs.AI]*, p. 1–55.

Participation de l'IRISA à DeFT 2018 : classification et annotation d'opinion dans des tweets

Anne-Lyse Minard[1] Christian Raymond[1,2] Vincent Claveau[1]
(1) CNRS, IRISA, Univ Rennes
(2) INSA Rennes, Rennes
Campus de Beaulieu, 35042 Rennes, France
prenom.nom@irisa.fr

RÉSUMÉ
Cet article décrit les systèmes développés par l'équipe LinkMedia de l'IRISA pour la campagne d'évaluation DeFT 2018 portant sur l'analyse d'opinion dans des tweets en français. L'équipe a participé à 3 des 4 tâches de la campagne : (i) classification des tweets selon s'ils concernent les transports ou non, (ii) classification des tweets selon leur polarité et (iii) annotation des marqueurs d'opinion et de l'objet à propos duquel est exprimée l'opinion. Nous avons utilisé un algorithme de boosting d'arbres de décision et des réseaux de neurones récurrents (RNN) pour traiter les tâches 1 et 2. Pour la tâche 3 nous avons expérimenté l'utilisation de réseaux de neurones récurrents associés à des CRF. Ces approches donnent des résultats proches, avec un léger avantage aux RNN, et ont permis d'être parmi les premiers classés pour chacune des tâches.

ABSTRACT
IRISA at DeFT 2018: classifying and tagging opinion in tweets

This paper describes the systems developed at IRISA by the LinkMedia team for the challenge DeFT 2018. The challenge focuses on opinion mining in French tweets about transports. The team has participated in 3 out of the 4 tasks: (i) classification of the tweets whether they are about transports or not, (ii) classification of the tweets according to their polarity and (iii) fine grained annotation of the sentiment expression and the object about which an opinion is given. For the tasks 1 and 2, we have used a boosting algorithm as well as recurrent neural networks (RNN). For the 3rd task, we have experimented the use of recurrent neural networks combined with some CRF. All the approaches give close results, with a slight advantage when using RNN, and yields among the best results for every tasks.

MOTS-CLÉS : analyse d'opinion, boosting, arbres de décision, réseau de neurones récurrents, plongement de mots, CRF.

KEYWORDS: opinion mining, boosting, decision trees, recurrent neural networks, word embedding, CRF.

1 Introduction

Cet article présente les systèmes que l'IRISA a développés dans le cadre de sa participation à la campagne d'évaluation DeFT 2018. Cette campagne porte sur l'analyse de sentiments dans des tweets qui concernent les transports. Elle fait suite à la campagne DeFT 2017 portant elle-aussi sur la

	# tweets	transport	positif	negatif	neutre	mixposneg
simple	54 638	35 468	7 328	13 109	12 611	2 420
batch_b	14 278	0	-	-	-	-
total	68 916	35 468	7 328	13 109	12 611	2 420

TABLE 1: Distribution des annotations dans le corpus d'entraînement pour les tâches 1 et 2.

fouille d'opinion. Nous avons donc repris et adapté certaines techniques développées précédemment (Claveau & Raymond, 2017). Notre équipe a participé à trois des quatre tâches proposées par les organisateurs :

— tâche 1 : classification des tweets selon qu'ils concernent les transports ou non ;
— tâche 2 : classification des tweets concernant les transports selon leur polarité (POSITIF, NEGATIF, NEUTRE, MIXPOSNEG) ;
— tâche 3 : identification des marqueurs de sentiments et de la cible du sentiment correspondant (annotation d'empans de texte et de relations).

La tâche 4, à laquelle nous n'avons pas participé, consiste à déterminer l'entité qui exprime le sentiment (source), les négations et les modifieurs, ainsi que les relations entre ces éléments.

Des données d'entraînement étant fournies par les organisateurs du challenge, nous avons classiquement adopté une approche d'apprentissage supervisé. Nous avons expérimenté deux méthodes d'apprentissage reposant sur des fondements différents, et sur des représentations différentes des données : le boosting d'arbres de décision et les réseaux de neurones récurrents.

Dans la suite de l'article, nous présentons dans la section 2 notre participation aux tâches 1 et 2, puis dans la section 3 nous détaillons nos expérimentations dans le cadre de la tâche 3.

2 Classification de tweets : tâche 1 et 2

La tâche 1 consiste à classer les tweets selon qu'il y soit question des transports ou non. Pour cela nous avons à disposition un corpus de 68 916 tweets dont 35 468 sont classés TRANSPORT (voir le tableau 1). Le corpus distribué par les organisateurs est composé de deux sous-corpus : "simple" et "batch_b". Dans la sous-partie "batch_b" du corpus, aucun tweet n'est classé TRANSPORT, même si un grand nombre de tweets concerne les transports. Nous avons donc décidé de n'utiliser que le sous-corpus "simple" pour entraîner nos modèles.

2.1 Bonzaiboost

Bonzaiboost est une implémentation de l'algorithme de boosting adaboost.MH (Laurent *et al.*, 2014) sur des arbres de décision. Cet algorithme est connu pour sa pertinence dans le domaine du traitement des langues et de l'apprentissage en général. Son utilisation constitue pour nous une solide référence sur laquelle se comparer afin d'expérimenter des systèmes plus sophistiqués. Cet algorithme appliqué sur des arbres de décision très faibles (2 feuilles) nous permet en outre de facilement interpréter le modèle appris et d'obtenir un retour d'information très intéressant. Son point faible, lié à l'algorithme de boosting lui-même, est de booster les exemples mal classés au long des itérations

lors de l'apprentissage : dans le cas de corpus bruités (avec la présence d'annotations erronées ou non cohérentes) l'algorithme insiste vainement à vouloir les classer. C'est notamment le cas, ici, où l'annotation d'opinions exprimées dans des tweets est relativement subjective et difficile.

Un modèle assez simple a été appris, où les caractéristiques extraites sont uniquement des sacs de mots convertis en minuscules. Le tableau 2 illustre les opinions marquées ainsi que la classe transport [1] par leur 13 règles les plus caractéristiques selon ce modèle. Afin de s'affranchir des variations orthographiques et de proposer des patrons plus généraux, nous avons testé d'apprendre un modèle sur des Ngrammes de caractères avec $N \in [3, 5]$ mais le système résultant est équivalent.

sncf	-2.849
@rera_ratp	-2.840
@rerc_sncf	-2.827
aéroport	-2.773
@rerb	-2.691
rer	-2.612
#sncf	-2.549
@sncf	-2.453
#ratp	-2.425
navigo	-2.277
trafic	-2.069
métro	-2.065
tramway	-2.048

(a) TRANSPORT

2016™	1.242
plaisir	1.149
ptdrrr	1.148
mdrrr	1.071
ptdrr	1.054
mdrr	1.042
mdrrrr	1.029
adore	1.020
sympa	1.000
beau	0.996
bravo	0.982
cool	0.980
rire	0.946

(b) POSITIVE

puent	1.423
fdp	1.161
accident	1.135
pue	1.112
gênant	1.074
pute	1.015
marre	0.993
flemme	0.989
honte	0.977
chier	0.939
merde	0.928
bordel	0.921
pire	0.899

(c) NÉGATIVE

TABLE 2: Treize mots les plus caractéristiques des tweets évoquant les TRANSPORTS et des opinions marquées : POSITIVES ou NÉGATIVES, accompagné de leur score de vote donné par l'algorithme de boosting.

2.2 BiLSTM+softmax

Pour résoudre ces deux tâches nous avons également expérimenté des approches à base de réseaux de neurones récurrents. La première méthode utilise une couche de LSTM bidirectionnelle (Graves *et al.*, 2013). La figure 1 décrit la méthode utilisée. La couche d'entrée prend une représentation des mots : concaténation des plongements des mots qui composent le tweet à classer et des one-hot vecteurs des mots (c'est-à-dire des vecteurs utilisés pour distinguer chaque mot dans un lexique). La couche de LSTM bidirectionnelle qui suit permet de prendre en compte l'aspect séquentiel des mots du tweet. Pour finir nous avons une couche cachée dense avec une activation softmax. En plus des couches décrites nous avons inséré des couches de Dropout pour éviter le sur-apprentissage. L'apprentissage est fait en 3 itérations et la taille du batch est celle par défaut, soit 32.

Les tweets sont prétraités de la façon suivante :

1. En l'occurrence pour la classe TRANSPORT, les mots caractéristiques sont ceux de l'absence de NON-TRANSPORT.

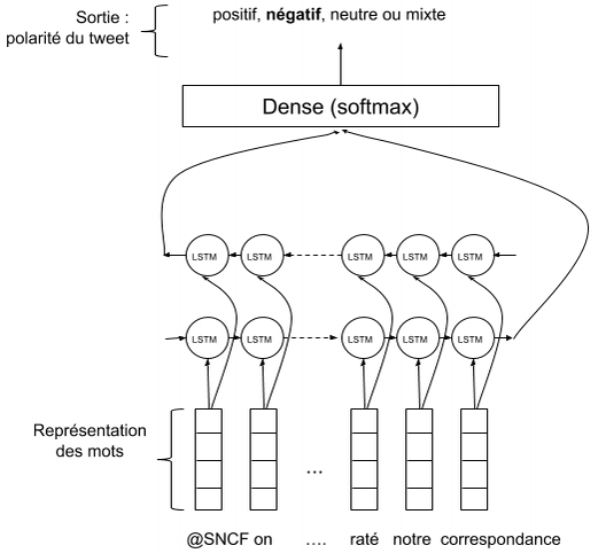


FIGURE 1: BiLSTM pour la classification des tweets.

— tokenization [2] avec le module tokenize de NLTK [3] ;
— remplacement des emojis par leur code [4] (par exemple l'emoji qui représente des mains qui applaudissent est remplacé par ":clap:") ;
— remplacement des nombres et des URLs respectivement par _num_ et _http_ ;
— tous les caractères sont convertis en minuscule ;
— tous les accents sont supprimés ainsi que la cédille.

Les plongements de mots utilisés pour représenter les mots du texte sont entraînés avec l'outil fasttext (Bojanowski *et al.*, 2017) sur des données de wikipedia et common crawl, et mis librement à disposition par Grave *et al.* (2018).

2.2.1 Expérience 1 : utilisation des données de DeFT 2017

La campagne d'évaluation DeFT 2017 s'est intéressée également à l'analyse de sentiment, avec un intérêt particulier porté aux tweets figuratifs. Les tâches 1 et 3 consistaient à classer les tweets selon leur polarité (objective, positif, négatif ou mixte). Pour la tâche 1, seuls les tweets non figuratifs étaient considérés et pour la tâche 3 à la fois des tweets figuratifs et non figuratifs. Nous avons évalué l'impact de l'utilisation des données d'entraînement et de test de DeFT 2017 sur la classification des tweets de DeFT 2018. Les résultats obtenus en validation croisée de 5 plis sont présentés dans le tableau 3. Nous observons qu'en utilisant les données de la tâche 1 de DeFT 2017 (3 906 tweets) en plus des données de DeFT 2018 nous n'améliorons pas les performances de notre système. Et

2. Les tweets fournis par les organisateurs avaient déjà été tokenisés, mais le module tokenize nous a permis d'améliorer le découpage.

3. `http://www.nltk.org/api/nltk.tokenize.html`

4. Pour remplacer les emojis par leur code nous utilisons le module python emoji disponible à l'adresse `https://github.com/carpedm20/emoji`.

DEFT 2018	+	+	+
DEFT 2017	-	T1	T3
POSITIF	64,92	64,98	63,11
NEGATIF	73,70	73,96	73,06
NEUTRE	74,24	74,07	73,70
MIXPOSNEG	31,37	31,58	25,40
Micro F-mesure	**69,93**	**69,94**	69,11

TABLE 3: Résultats obtenus en utilisant différents ensembles de données d'entraînement pour la tâche 2.

en utilisant les données de la tâche 3 de DeFT 2017 (5 118 tweets) les performances diminuent légèrement.

Les deux corpus contiennent des tweets sur des sujets différents : sujets d'actualité pour DeFT 2017 et transports pour DeFT 2018. Cette différence peut expliquer pourquoi l'utilisation des données de DeFT 2017 ne permet pas d'améliorer les performances de notre classifieur. En particulier en observant les mots du tableau 2 et ceux du tableau 5 dans l'article de Claveau & Raymond (2017), nous remarquons une grande différence dans le type des mots exprimant des opinions (par exemple pour la polarité négative "puent", "fdp", "accident", "pue" versus "pauvre", "nul", "sarko", "plein", "gvt").

Dans la suite de nos expériences nous avons donc utilisé uniquement les tweets de DeFT 2018.

2.2.2 Expérience 2 : variation de la quantité des données d'apprentissage utilisées

Nous nous sommes également intéressés aux performances de notre système en fonction de la quantité de données d'apprentissage utilisées. La figure 2 présente l'évolution de la micro F-mesure (générale et pour chaque valeur de polarité) en fonction du nombre de données d'apprentissage. Nous observons que les performances maximales du système sont atteintes avec 80% du jeu d'entraînement (soit environ 28 400 tweets). Nous pouvons faire les mêmes observations pour les classes POSITIF, NÉGATIF et NEUTRE. En revanche pour la classe MIXPOSNEG les performances augmentent encore lorsque la totalité des données d'entraînement est utilisée. Cette évaluation a été faite en validation croisée de 5 plis.

2.3 Variantes autour des RNN

Sur la base de l'approche précédente, nous avons exploré de nombreuses variantes de structure du réseau. Nous avons en particulier étudié l'apport de modèles d'attention. Ces modèles, très populaires, permettent dans le cas de données séquentielles comme le sont nos tweets, de fonder la décision du réseau sur la base de certains mots. C'est-à-dire que le réseau va être entraîné à donner beaucoup de poids aux mots de l'entrée pertinents pour prédire la classe attendue, et très peu de poids aux autres mots. En pratique, ces modèles d'attention sont implémentés sous la forme d'une couche de neurones supplémentaire avec une activation softmax et dont les poids sont ensuite multipliés à la sortie de la couche BiLSTM (ou BiGRU).

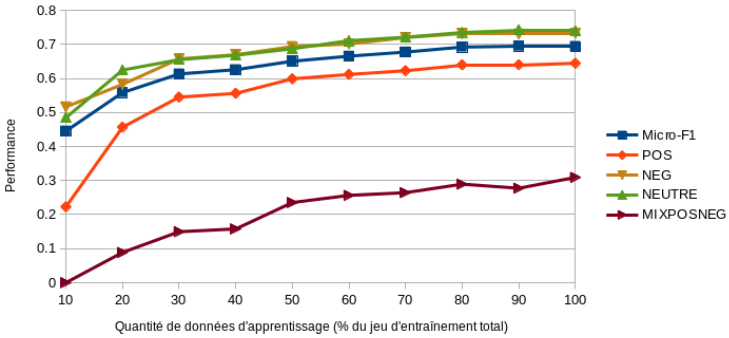


FIGURE 2: Évolution des performances du système en fonction du nombre de données d'apprentissage utilisées.

Nous avons ainsi proposé cette simple variante du système de la sous-section précédente pour la tâche 1. Pour limiter les effets du sur-apprentissage, ces modèles sont appris avec peu d'itérations, nous avons proposés un *run* avec un arrêt après trois itérations (noté RNN3) et un autre après cinq itérations (RNN5).

Pour la tâche 2, nous avons modifié plus profondément le réseau présenté dans la sous-section précédente. Nous avons deux branches avec la même architecture (Embedding, Bi-LSTM, couche d'attention, et un neurone de sortie par branche). Une branche apprend si le tweet est POSITIF ou non, et l'autre s'il est NEGATIF ou non. La combinaison des deux sorties permet bien d'avoir les quatre classes possible (NEUTRE quand les deux branches renvoient 0, MIXPOSNEG quand les deux renvoient 1). Là encore, l'apprentissage des modèles est limité à 3 et 5 itérations.

Enfin, d'autres variantes ont également été testées, portant soit sur la préparation des données, soient sur l'architecture du réseau. Leurs résultats en validation croisée étant identiques aux approches décrites ci-dessus, elles n'ont pas été soumises à l'évaluation finale. Nous avons par exemple étudié l'impact des données textuelles utilisées pour apprendre les plongements de mots, sans constater de différences importantes entre des tweets et du texte propre (wikipedia, journaux). Nous avons également testé des implémentations de word2vec et fasttext avec différents paramètres. Nous avons également essayé de diminuer l'importance des mots thématiques pour la tâche 2. Pour cela nous avons récupéré les poids de ces mots dans la couche d'attention du réseau utilisé pour la tâche 1. En effet, ces mots marqueurs thématiques du transport reçoivent un poids important pour la tâche 1. En inversant ces poids et en les intégrant à une couche, que l'on pourrait appelé couche d'inattention, dans le réseau de la tâche 2, nous espérions que le réseau se focaliserait sur les mots marqueurs d'opinion. Mais nous n'avons pas constaté de différences importantes, pour un réseau plus complexe et plus long à entraîner.

2.4 Résultats

Dans le tableau 4, nous présentons les résultats obtenus par nos 4 systèmes pour les tâches 1 et 2. Ces résultats ont été fournis par les organisateurs de la campagne d'évaluation.

En entrée des systèmes pour la tâche 2 nous utilisons les données obtenues en sortie de la tâche 1.

		Tâche 1	Tâche 2		
		P	P	R	F
Bonzaiboost	run2	82,06	67,95	94,14	78,93
BiLSTM+softmax	run1	82,66	70,23	96,50	**81,31**
RNN3	run3	82,51	69,82	95,66	80,72
RNN5	run4	**82,70**	69,81	96,23	80,92

TABLE 4: Résultats obtenus par nos systèmes pour les tâches 1 et 2 de la campagne d'évaluation DeFT2018.

Les tweets classés par erreur comme TRANSPORT (faux positif) sont ignorés lors de l'évaluation de la tâche 2. En revanche les tweets qui concernent les transports et qui ont été classés par erreur dans la classe NON-TRANSPORT (faux négatif) sont comptés comme faux négatif dans la tâche 2.

Pour la tâche 1, notre meilleur système (RNN5) obtient une précision de 82,70%, 0,42 points en dessous du meilleur système de la compétition (différence non statistiquement significative d'après un t-test pairé avec p=0.05). Pour la tâche 2, les meilleures performances sont obtenues par le système BiLSTM+softmax avec une F-mesure de 81,31%, 0,98 points de moins que le meilleur système de la compétition.

3 Annotation fine d'opinion/sentiment/émotion : tâche 3

La tâche 3 consiste à annoter les empans de texte exprimant une opinion, un sentiment ou une émotion, ainsi que la cible du sentiment, c'est-à-dire l'objet [5] à propos duquel est exprimée une opinion. Les marqueurs qui expriment un sentiment, une émotion ou une opinion (OSEE) sont associés à 20 types différents. Dans le tableau 5, nous présentons les différents types de marqueurs et leur distribution dans les données d'entraînement. Il nous était ensuite demandé de relier la cible à l'expression de l'opinion.

Cette tâche se rapproche de la tâche "Aspect-Based Sentiment Analysis" qui a eu lieu pour la première fois à SemEval 2014 (Pontiki *et al.*, 2014). Elle consistait en quatre sous-tâches, les deux premières étant les plus proches de la tâche 3 de DeFT2018 : (i) extraction de l'aspect, c'est-à-dire l'attribut de l'objet sur lequel une opinion est donnée ; (ii) identification de la polarité associée à l'aspect. À SemEval 2016 (Pontiki *et al.*, 2016), la tâche est devenue multilingue avec entre autres des données pour le français (Apidianaki *et al.*, 2016). Les deux meilleurs systèmes pour le français étaient basés sur des CRF (Conditional Random Field) (Brun *et al.*, 2016; Kumar *et al.*, 2016). Pour l'anglais, Toh & Su (2016) ont expérimenté l'utilisation de réseaux de neurones récurrents (bidirectionnel Elman-type RNN) associés à des CRF et obtiennent les meilleurs résultats de la tâche "Aspect-Based Sentiment Analysis" de SemEval 2016. Ils ont observé que l'utilisation d'un Elman-type RNN bidirectionnel associé à des CRF améliore les performances du système par rapport à des CRF seuls. L'association d'une couche de RNN bidirectionnel et de CRF a été testé sur plusieurs tâches de *sequence labelling* ces dernières années et a souvent permis de dépasser les résultats état-de-l'art. Les BiLSTM et BiGRU sont souvent employés (Huang *et al.*, 2015; Ma & Hovy, 2016; Dalloux *et al.*, 2017).

5. Le terme "objet" utilisé par les organisateurs est à prendre au sens large, en effet il inclut également des situations ou événements, des personnes, etc.

	NEGATIF		POSITIF	
émotion	déplaisir	631	plaisir	4 919
	dérangement	2 061	apaisement	541
	mépris	2 359	amour	560
	surprise négative	140	surprise positive	118
	peur	961		
	colère	2 090		
	ennui	89		
	tristesse	1 042		
sentiment	insatisfaction	1 484	satisfaction	1 991
opinion	désaccord	1 785	accord	580
	dévalorisation	2 826	valorisation	6 982
type générique	négatif	9 949	positif	2 838

TABLE 5: Distribution des types de marqueurs d'opinion, sentiment et émotion dans le corpus d'entraînement.

Nous avons traité cette tâche comme une tâche de *sequence labeling*. Nous avons expérimenté une méthode basée sur des réseaux de neurones récurrents et des CRF. Les relations entre la cible et l'OSEE ont été extraites avec une simple règle de proximité. Dans cette partie, nous décrivons dans un premier temps le corpus, puis la méthode utilisée, les expériences effectuées et enfin les résultats obtenus.

3.1 Corpus

Les données distribuées par les organisateurs contiennent 68 916 tweets (voir tableau 1). Pour 44 742 de ces tweets, une annotation pour la tâche 3 est disponible. Pour cette tâche sont considérés uniquement les tweets qui concernent les transports et qui ont une polarité positive, négative ou mixte, ce qui ne concernent en théorie que 22 857 tweets. Nous avons donc à disposition plus d'annotations que celles répondant aux critères de la tâche 3. Dans le tableau 6, nous présentons le nombre d'annotations de CIBLE (objet à propos duquel est exprimée une opinion) et de OSEE (expression d'opinion, sentiment et émotion) pour différents sous-corpus. La colonne "distribué" indique le nombre total d'annotations disponibles. La colonne "transport" contient les informations concernant les tweets TRANSPORT [6], quelque soit leur polarité. La colonne "POS/NEG" indique le nombre d'annotations disponibles dans l'ensemble des tweets répondant aux critères de la tâche.

Nous avons converti les données au format IOB2 [7]. Malheureusement certains offsets étaient erronés et nous n'avons donc pu utiliser que 89% du corpus avec une incertitude sur la qualité des données. Nous donnons ci-dessous un exemple de tweet annoté avec les offsets fournis et qui illustre le problème rencontré :

<valorisation>Fort de</valorisation> mon <positif>talent</positif> <source>, j</source>' a<insatisfaction>i ra</insatisfaction>t<cible>é le voyage en bu</cible>s.

6. Pour le sous-corpus "batch_b", nous avons effectué une classification automatique des tweets pour distinguer ceux qui concernent les transports des autres.

7. Dans le format IOB2, B- indique le début d'un chunk, I- indique qu'un token est à l'intérieur d'un chunk et O qu'un token ne fait pas partie d'un chunk.

	distribué			transport			POS/NEG		
	CIBLE	OSEE	rel	CIBLE	OSEE	rel	CIBLE	OSEE	rel
simple	34 772	43 946	43 641	30 400	38 622	38 218	26 198	36 563	32907
simple IOB2	30 488	62 214	-	30 488	62 214	-	23 063	51 006	-
batch_b	83 087	48 425	0	59 033	30 198	0	-	-	-
batch_b IOB2	69 969	72 129	-	47 722	45 556	-	-	-	-

TABLE 6: Distribution des annotations dans le corpus d'entraînement pour la tâche 3.

Les OSEE sont classifiées en 20 classes (voir tableau 5). Dans le cas où il serait difficile de classer une expression dans une des 18 premières classes, le marqueur est typé uniquement comme "négatif" ou "positif". Nous remarquons qu'un grand nombre de marqueurs est associé à une de ces deux classes (30%).

3.2 Méthodes

Pour résoudre cette tâche nous avons expérimenté une méthode basée sur des réseaux de neurones récurrents et des CRF (Lafferty *et al.*, 2001). La méthode est illustrée dans la figure 3. Chaque mot du tweet est représenté par une concaténation d'un vecteur issu d'un plongement et d'un one-hot vecteur. Ces vecteurs sont fournis en entrée à une couche de GRU bidirectionnelle (BiGRU) (Cho *et al.*, 2014). La couche de sortie est une couche de CRF qui prédit de façon séquentielle les étiquettes des mots du tweet. Des couches Dropout sont ajoutées pour éviter le sur-apprentissage. L'apprentissage est fait en 3 itérations et la taille du batch est celle par défaut, soit 32.

Dans la phase d'expérimentation nous avons testé à la fois l'utilisation de BiLSTM et de BiGRU. Les résultats obtenus avec une couche BiGRU étaient légèrement meilleurs qu'avec une couche BiLSTM.

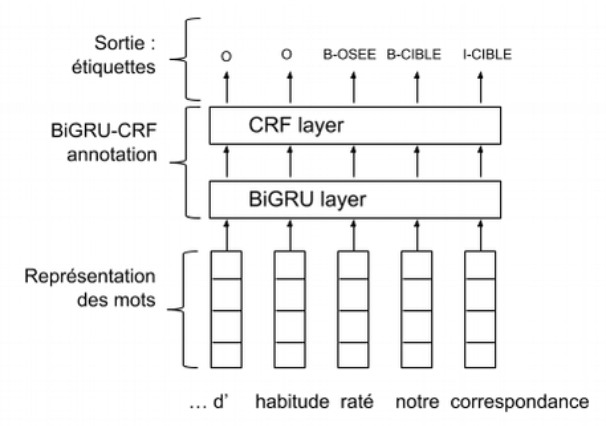


FIGURE 3: BiGRU-CRF pour l'annotation des expressions de sentiment et des cibles.

Nous avons effectué les prétraitements suivants sur les données :

— remplacement des nombres, des URLs, des hashtags et des alias respectivement par _num_, _http_, _hashtag_ et _alias_ ;

— tous les caractères sont convertis en minuscule ;

— tous les accents sont supprimés ainsi que la cédille.

Les plongements de mots utilisés ont été appris avec fasttext sur une corpus composé de wikipedia et common crawl Grave *et al.* (2018).

Les relations entre les cibles et les OSEE sont extraites en utilisant une simple règle de proximité, c'est-à-dire qu'une cible est reliée aux marqueurs d'opinion les plus proches en termes de nombre de mots. Nous avons défini cette règle après observation d'exemples du corpus mais n'avons pas réalisé d'expériences pour vérifier sa validité.

3.3 Expériences

Nous avons expérimenté trois approches qui se différencient par le nombre de labels appris par un modèle et le nombre de modèles :

— **2 branches** : un modèle entraîné pour l'annotation des cibles (B-CIBLE, I-CIBLE et O) et un pour l'annotation des OSEE (41 étiquettes : O, B-PEUR, I-PEUR, B-AMOUR, etc.) ;
— **2 étapes** : un modèle entraîné pour annoter les cibles et les OSEE sans y associer un type précis (B-CIBLE, I-CIBLE, B-OSEE, I-OSEE et O) et un pour typer les OSEE (41 étiquettes) ;
— **1 étape** : un modèle entraîné pour l'annotation des 43 étiquettes.

	CIBLE	OSEE
2 branches	31,19	58,12
2 étapes	29,50	57,71
1 étape	27,47	57,24

TABLE 7: Évaluation des trois approches proposées (en termes de micro F-mesure).

L'évaluation a été effectuée en utilisant le scorer de CoNLL 2003. Une séquence est considérée comme correcte si son empan correspond exactement à un empan dans l'annotation de référence (strict match) et que l'étiquette qui leur est associée est identique. L'évaluation est faite en validation croisée à 5 plis sur les données décrites dans la colonne "POS/NEG" du tableau 6. Les résultats sont donnés dans le tableau 7 en termes de micro F-mesure. Les meilleures performances sont obtenues avec la méthode "2 branches" avec une mico F-mesure de 31,19% pour l'annotation de la cible et de 58,12% pour l'annotation et la classification des OSEE.

	simple			simple POS/NEG			simple + batch_b		
	P	R	F	P	R	F	P	R	F
CIBLE	42,03	26,55	**32,51**	38,34	26,41	31,19	25,30	22,70	23,32
OSEE	64,01	54,70	**58,97**	61,31	55,27	58,12	51,94	53,32	52,60

TABLE 8: Évaluation de l'impact du corpus d'entraînement utilisé avec l'approche "2 branches" en termes de précision (P), rappel (R) et F-mesure (F).

Dans la section 3.1, nous avons décrit le corpus et indiqué que les données distribuées contenaient plus de tweets annotés que ceux répondant aux critères de la tâche. Pour évaluer l'impact de l'utilisation de ces données disponibles, nous avons entraîné notre meilleur système ("2 branches") avec différents sets de données d'entraînement. Dans le tableau 8, nous présentons les résultats obtenus en validation croisée à 5 plis. Nous observons que les meilleures performances sont atteintes en utilisant le sous-corpus "simple" en entier (c'est-à-dire à la fois les tweets positifs, négatifs mixtes et les tweets

		micro F-mesure	CIBLE	OSEE
2 branches	run1	40,00	26,25	49,64
2 étapes	run2	39,69	23,11	51,43
1 étape	run3	44,02	22,63	56,24

TABLE 9: Résultats obtenus par nos systèmes pour la tâche 3 de la campagne d'évaluation DeFT2018 (résultats non officiels).

neutres). Nous remarquons que lorsque nous utilisons le corpus "batch_b" en plus du corpus "simple" la précision chute de plus de 10 points, ce qui montre la basse qualité de ces annotations.

3.4 Résultats

Nous présentons dans cette partie les résultats obtenus pour la tâche 3 sur les données d'évaluation. Les résultats obtenus pour chaque run sont présentés dans le tableau 9. Ces scores ont été calculés par nos soins, en utilisant le scorer de CoNLL 2003. Aucune évaluation des relations entre cible et marqueur de sentiment n'a été faite. Comme pour les évaluations présentées dans la sous-section précédente, la comparaison entre la référence et la sortie du système est faite en "strict match". L'évaluation ne porte que sur les tweets annotés à la fois dans la référence et dans la sortie du système.

Les modèles ont été entraînés en utilisant toutes les données disponibles ("simple + batch_b"). Nous avons montré dans la sous-section précédente que l'utilisation des données du sous-corpus "batch_b" faisait diminuer les performances du système, mais cette observation a été faite après la phase d'évaluation.

Dans le tableau 9, nous observons que les meilleures performances pour la détection et la classification des marqueurs de sentiment sont obtenues avec l'approche "1 étape" avec une F-mesure de 56,24, contrairement aux résultats obtenus en validation croisée sur les données d'entraînement. Pour l'annotation des cibles à propos desquelles une opinion est exprimée, l'approche permettant d'obtenir les meilleurs résultats est l'approche "2 branches" avec une F-mesure de 26,25, performances qui restent très basses.

4 Conclusion

Nous avons présenté dans cet article la participation de l'équipe LinkMedia de l'IRISA à la campagne d'évaluation DeFT 2018. Nous avons développé des systèmes basés sur le boosting d'arbres de décisions et sur des réseaux de neurones récurrents pour les deux premières tâches. Nous avons observé que l'algorithme de boosting obtient des résultats comparables aux RNN pour la tâche 1 de classification des tweets en TRANSPORT/NON-TRANSPORT. En revanche pour la tâche 2, classification selon la polarité, les RNN obtiennent des meilleures performances.

Pour la tâche 3, que nous avons traité comme une tâche de *sequence labelling*, nous avons développé une méthode à base de RNN et CRF. Nous sommes les seuls participants de cette tâche, nous n'avons donc pas de points de comparaison avec d'autres approches. Nous espérons pouvoir très prochainement entraîner de nouveau nos modèles sur des données propres (c'est-à-dire pour lesquelles les offsets ont été corrigés).

Références

APIDIANAKI M., TANNIER X. & RICHART C. (2016). Datasets for Aspect-Based Sentiment Analysis in French. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* : European Language Resources Association (ELRA).

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

BRUN C., PEREZ J. & ROUX C. (2016). XRCE at SemEval-2016 Task 5 : Feedbacked Ensemble Modeling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 277–281, San Diego, California : Association for Computational Linguistics.

CHO K., VAN MERRIENBOER B., BAHDANAU D. & BENGIO Y. (2014). On the Properties of Neural Machine Translation : Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, p. 103–111 : Association for Computational Linguistics.

CLAVEAU V. & RAYMOND C. (2017). IRISA at DeFT2017 : classification systems of increasing complexity . In *DeFT 2017 - Défi Fouille de texte*, Actes de l'Atelier Défi Fouille de Texte, DeFT, p. 1–10, Orléans, France.

DALLOUX C., CLAVEAU V. & GRABAR N. (2017). Détection de la négation : corpus français et apprentissage supervisé. In *SIIM 2017 - Symposium sur l'Ingénierie de l'Information Médicale*, p. 1–8, Toulouse, France.

GRAVE E., BOJANOWSKI P., GUPTA P., JOULIN A. & MIKOLOV T. (2018). Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

GRAVES A., MOHAMED A.-R. & HINTON G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 6645–6649 : IEEE.

HUANG Z., XU W. & YU K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR*, **abs/1508.01991**.

KUMAR A., KOHAIL S., KUMAR A., EKBAL A. & BIEMANN C. (2016). IIT-TUDA at SemEval-2016 Task 5 : Beyond Sentiment Lexicon : Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 1129–1135, San Diego, California : Association for Computational Linguistics.

LAFFERTY J., MCCALLUM A. & PEREIRA F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, p. 282–289 : Morgan Kaufmann, San Francisco, CA.

LAURENT A., CAMELIN N. & RAYMOND C. (2014). Boosting bonsai trees for efficient features combination : application to speaker role identification. In *InterSpeech*, Singapour.

MA X. & HOVY E. H. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 1064–1074, Berlin, Germany : Association for Computational Linguistics.

PONTIKI M., GALANIS D., PAPAGEORGIOU H., ANDROUTSOPOULOS I., MANANDHAR S., AL-SMADI M., AL-AYYOUB M., ZHAO Y., QIN B., DE CLERCQ O., HOSTE V., APIDIANAKI

M., TANNIER X., LOUKACHEVITCH N., KOTELNIKOV E., BEL N., JIMÉNEZ-ZAFRA S. M. & ERYIĞIT G. (2016). SemEval-2016 Task 5 : Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 19–30, San Diego, California : Association for Computational Linguistics.

PONTIKI M., GALANIS D., PAVLOPOULOS J., PAPAGEORGIOU H., ANDROUTSOPOULOS I. & MANANDHAR S. (2014). SemEval-2014 Task 4 : Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 27–35, Dublin, Ireland : Association for Computational Linguistics and Dublin City University.

TOH Z. & SU J. (2016). NLANGP at SemEval-2016 Task 5 : Improving Aspect Based Sentiment Analysis using Neural Network Features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 282–288, San Diego, California : Association for Computational Linguistics.

DEFT 2018: Attention sélective pour classification de microblogs

Charles-Emmanuel Dias[1] Clara Gainon de Forsan de Gabriac[1] Vincent Guigue[1]
Patrick Gallinari[1]
Sorbonne Université, CNRS,
Laboratoire d'Informatique de Paris 6, LIP6,
F-75005 Paris, France
<prenom.nom-de-famille>@lip6.fr

RÉSUMÉ

Dans le cadre de l'atelier DEFT 2018 nous nous sommes intéressés à la classification de microblogs (ici, des tweets) rédigés en français. Ici, nous proposons une méthode se basant sur un réseau hiérarchique de neurones récurrent avec attention. La spécificité de notre architecture est de prendre en compte –via un mechanisme d'attention et de portes– les *hashtags* et les mentions directes (*e.g.,* @*user*), spécifiques aux microblogs. Notre modèle a obtenu de très bon résultats sur la première tâche et des résultats compétitifs sur la seconde.

ABSTRACT

DEFT 2018 : Selective Attention for Microblogging Classification

The 2018 DEFT challenge allowed us to investigate sentiment analysis and document classification applied to microblogs (here, twitter) written in French. We hereby present a method based on a hierarchical attentional recurrent neural network. Our architecture is specifically engineered to take advantage of hashtags and direct mentions – specificic of microblogs – by the mean of an attention and gate mechanism. Our model scored very good results on the first task and competitve ones on the second task.

MOTS-CLÉS : Classification, Analyse de Sentiments, Réseaux de Neurones, Attention..

KEYWORDS: Classification, Sentiment Analysis, Neural Networks, Attention..

1 Introduction

L'atelier DEFT 2018 (Paroubek *et al.*, 2018) proposait différentes tâches autours de l'extraction d'informations de tweets francophones. Ici, nous nous sommes focalisés sur les deux premières tâches de classification. Pour résoudre ces deux problèmes, nous proposons ici de nous inspirer d'un modèle neuronal hiérarchique issu de l'analyse de sentiments (Yang *et al.*, 2016) que nous modifions pour proposer un modèle de représentation prenant en compte les diverses spécificités des microblogs.

En effet, les tweets ont souvent une orthographe approximative (lettres répétées, ponctuation excessive ou manquante, smileys...), un vocabulaire spécifique (acronymes, mentions, hashtags...) et même des lettres particulières (emojis en unicode). Aussi, ils regorgent de mots-clés (dits *hashtags*) et de mentions directes (*@user*). Ces mots spécifiques, précédés de marqueurs (# et @), ont souvent une forte valeur informative qu'il convient de proprement intégrer au sein du modèle d'encodage. Pour

toutes ces raisons, représenter efficacement un tweet est particulièrement compliqué, surtout sur des petits corpus. C'est généralement au prix d'une multitude d'heuristiques complexes qu'il est possible de dépasser toutes ces contraintes pour atteindre des performances proches de celles des modèles neuronaux. À l'inverse, en apprenant des représentations latentes du texte (ou en utilisant des représentation pré-apprises (Mikolov *et al.*, 2013; Pennington *et al.*, 2014)), les réseaux de neurones profonds sont moins impactés par ces problèmes syntaxiques et sémantiques. Ils sont capables –sans aucun pré-traitement– de détecter des constructions textuelles avancées (comme les doubles négations) et peuvent même encoder une certaine information sémantique (pluriel, féminin-masculin,...). De ce fait, les machines à vecteur de support se basant sur des représentations de type *sac de mots* ou de *N-grams*, qui furent longtemps les modèles les plus performants pour résoudre les tâches de classification de documents, sont désormais majoritairement surpassés par les modèles neuronaux avec leurs représentations continues.

Ici, nous nous inspirons des travaux de (Yang *et al.*, 2016) pour construire notre modèle de représentation et de classification de microblogs. Ils proposent d'encoder hiérarchiquement le texte (mot par mot puis phrase par phrase) tout en apprenant conjointement –via un mécanisme d'attention– quels sont les éléments discriminants. Contrairement à eux, nous faisons l'hypothèse qu'encoder les tweets caractère par caractère est plus pertinent que mot à mot. Nous postulons qu'un tel encodage permet de nous abstraire de chacun des problèmes grammaticaux énoncés précédemment. Aussi, pour prendre en compte les hashtags et les mentions, nous y ajoutons un mécanisme d'attention et d'interpolation spécifique.

Cet article est organisé de la façon suivante : nous détaillons dans un premier temps les principaux éléments de notre modèle avant de présenter son architecture globale (section 2). Ensuite, nous évaluons notre modèle quantitativement et commentons les résultats obtenus lors de la phase d'évaluation (section 3). Enfin, nous proposons certaines pistes pour l'amélioration de notre méthode de représentation de tweet (section 4).

2 Modèle de classification de microblogs

Généralement, un tweet est un texte court –souvent écrit sur mobile– dont l'orthographe peut être approximative. Ces textes contiennent souvent des mots inexistants, abrégés ou argotiques. Nous considérons que seul l'espace est un caractère fiable, et qu'il agit comme un séparateur entre les mots. Aussi, nous faisons le choix de ne pas prendre les phrases en compte mais uniquement les caractères et les mots (séparés par les espaces). Ici, nous voyons donc un tweet comme une suite de caractères, divisée en plusieurs mots.

Pour notre modélisation nous nous inspirons du modèle hiérarchique d'analyse de sentiment de (Yang *et al.*, 2016) qui est composé de deux sous-entités similaires : des modules bi-directionnels attentifs (nous les appelons RBA). Leur rôle est d'encoder tour à tour les mots et les phrases pour représenter un texte avant d'en prédire sa polarité. Ici, ne prenant pas en compte les phrases, nous descendons d'un niveau hiérarchique et nous proposons d'encoder séquentiellement les mots –caractère par caractère– puis le message mot par mot.

Dans un premier temps nous explicitons la construction d'un module bi-directionnel attentif. Ensuite, nous détaillons comment le modèle prend en compte les mots clés et les mentions avant de décrire l'intégralité du modèle.

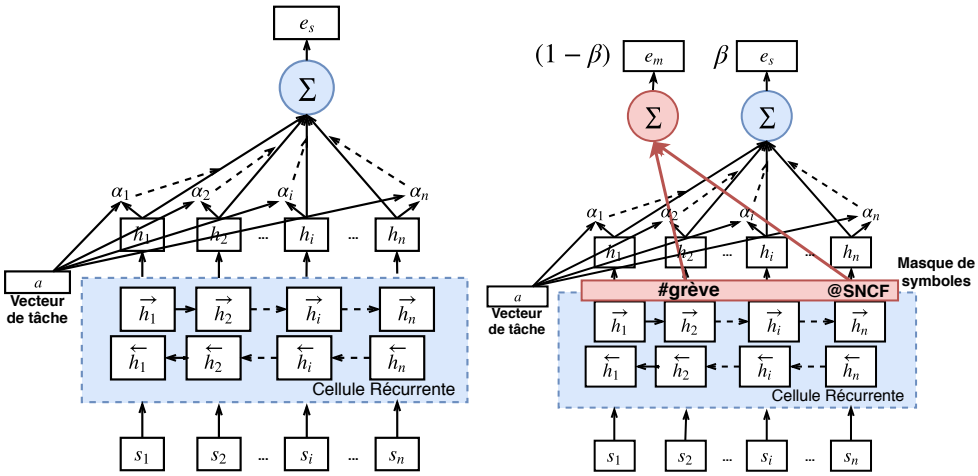


FIGURE 1 – Module **R**écurrent **B**i-directionnel avec **A**ttention **(RBA)** : Sans attention personnalisé (gauche) – Avec attention personnalisé (droite)

2.1 Module Récurrent Bi-directionel avec attention : le RBA

Ce module est le principal bloc de notre modèle de classification. Il prend en entrée une séquence et retourne une représentation transformée et pondérée de celle-ci. Dans le reste de l'article, nous appelons ces sous-modules RBA pour *Module **R**eccurent **B**i-directionnel avec **A**ttention*.

Encodage d'une séquence avec un RBA

Formellement, soit une séquence $seq = \{\mathbf{s}_1, \ldots, \mathbf{s}_i, \ldots, \mathbf{s}_n\}$ composée de n éléments. Pour obtenir sa représentation $\mathbf{e}_s$, la séquence est d'abord passée par un réseau de neurones récurrent bi-directionnel $RB = \{\overrightarrow{RB}, \overleftarrow{RB}\}$ qui, en parcourant la séquence dans les deux sens, encode le contenu intra-séquence. Les sorties du réseau récurrent sont concaténées à chaque pas de temps pour obtenir l'ensemble des représentations cachées $\mathbf{h}_i$ (eq. 1). Ici, nous utilisons une cellule GRU (Chung *et al.*, 2014) comme cellule récurrente.

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}], \qquad \overrightarrow{\mathbf{h}_i} = \overrightarrow{RB}(\mathbf{s}_i), \qquad \overleftarrow{\mathbf{h}_i} = \overleftarrow{RB}(\mathbf{s}_i), \tag{1}$$

Ensuite, chaque élément $\mathbf{h}_i$ est projeté de manière non linéaire dans un espace d'attention afin de calculer son affinité α_i avec un vecteur de tâche **a** –qui est lui-même appris lors de l'optimisation– selon la formule suivante :

$$\mathbf{t}_i = \tanh(W^{tu}\mathbf{h}_i + b_u), \qquad \alpha_i = \frac{\exp(\mathbf{a}^\intercal \mathbf{t}_i)}{\sum_i \exp(\mathbf{a}^\intercal \mathbf{t}_i)} \tag{2}$$

Ces affinités α_i sont normalisées à l'aide d'une fonction *softmax* afin qu'elles somment à 1. Ce vecteur de tâche **a** correspond au point optimal dans l'espace d'attention. (Yang *et al.*, 2016) a

montré qu'un tel vecteur de tâche –appris comme un paramètre– permettait au modèle de se focaliser automatiquement sur les éléments discriminants d'une séquence en fonction de la tâche.

$$\mathbf{e}_s = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i \tag{3}$$

Enfin, la représentation finale e_s de la séquence d'entrée est la somme des représentations cachés h_i, pondérée par l'attention α.

Attention personnalisé sur les mots-clés #hastags et les mentions @users

Si l'idée de pondérer les éléments d'une séquence pour dénoter leur importance relative est très répandue, l'originalité de notre approche réside dans une prises en compte particulière des éléments propres aux tweets : les hashtags et les mentions. En effet, nous partons du postulat que ces deux éléments fournissent des indices important pour les tâches subséquentes de classification.

Pour mieux prendre en compte ces mots spécifiques, nous proposons un système d'ajouter un système d'attention-interpolation au RBA qui servira à encoder les représentations de mots en une représentation du tweet. Nous travaillons donc sur le second niveau hiérarchique.

Nous proposons dans un premier temps de sélectionner les termes importants via un processus de masquage : soit la matrice –encodant m mots sur n dimensions– de l'ensemble des états cachés $H \in \mathcal{R}^{m,n}$ et $\mathbf{M} \in \{0^n, 1^n\}^m$ une matrice avec $m_{i,*} = 1^n$ si le i-eme mot commence par un caractère prédéfini –ici le dièse # ou l'arobase @– et $m_{i,*} = 0^n$ sinon. La représentation de ces mots spécifique est simplement leurs somme. $\otimes$ est le produit terme à terme.

$$e_m = \sum_i (H \otimes M) \tag{4}$$

Finalement, la représentation du tweet e_t est l'interpolation entre cette somme de mots sélectionnés e_m et la somme pondérée par l'attention classique e_s obtenue sur l'intégralité des mots. Cette interpolation est linéaire, de coefficient β, fonction des deux représentations.

$$e_t = (1 - \beta) \times e_m + \beta \times e_s, \qquad \beta = \sigma(W^b[e_s; e_t] + b_b), \qquad \sigma = sigmoid(x) \tag{5}$$

Ce système d'attention permet de sur-pondérer les termes importants dans la représentation finale. Cela permet à l'attention classique de se focaliser sur les marqueurs traditionnels.

2.2 Architecture globale du modèle

Notre modèle fonctionne de manière hiérarchique. Il prend en entrée une liste de liste de caractères et prédit une classe. Tout d'abord, n représentations latentes de mots $e_w(k)$ sont construites au moyen d'un premier RBA (RBA_c) qui encode chaque k mot de m lettres caractère par caractère. Ensuite, à l'aide d'un second RBA (RBA_m), la représentation finale du tweet e_t est construite à partir toutes les représentations de mots précédemment obtenues.

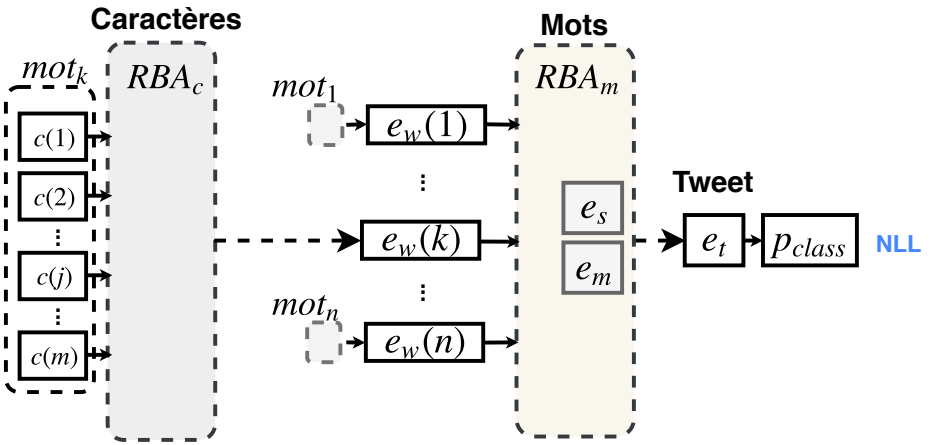


FIGURE 2 – Modèle hiérarchique de classification de microblog complet, composé de deux RBA. RBA_w encode d'abord chaque mots caractère par caractère puis RBA_m encode le tweet mot à mot en utilisant le système d'attention personnalisée

$$e_w(k) = RBA_c([c_0, \cdots, c_m]), \qquad e_t = RBA_m([e_w(1), \cdots, e_w(k), \cdots, e_w(n)]) \tag{6}$$

Enfin, une couche de classification softmax permet la classification finale.

$$p_{class} = softmax(W^{tc} e_t + b_c) \tag{7}$$

Pour entraîner le modèle, nous minimisons l'entropie croisée par descente de gradient par mini-batch. Ici, nous proposons le même modèle pour les deux tâches, la taille de la couche finale varie donc en fonction du nombre de classes à prédire.

3 Evaluation

Dans le cadre de cet atelier, nous présentons une modification du modèle de (Yang *et al.*, 2016) prenant en compte certaines spécificités des microblogs. Proposant un modèle de classification, nous participons aux tâches 1 et 2. L'une étant bi-classes et l'autre multi-classes.

Dans cette section, nous présentons dans un premier temps les données ainsi que les deux tâches que nous avons considérés. Puis, nous reportons les évaluations de notre modèle. Enfin, nous commentons les résultats obtenus.

3.1 Données, pré-processing et tâches

Les données de l'atelier DEFT2018 (Paroubek *et al.*, 2018) sont –modulo les erreurs d'annotation– des tweets annotés selon leur polarité si et seulement s'ils traitent des transports. Les étiquettes sont

aux nombre de cinq : INCONNU, NEUTRE, NEGATIF, POSITIF, MIXPOSNEG.

Tâche 1 – Transport/non-transport : Cette première tâche consiste à faire de la classification thématique. L'enjeu est de séparer les tweet traitant du transport de ceux parlant d'autres choses. Formellement, les tweets avec l'étiquette INCONNU sont considéré comme ne traitant pas de transports alors que ceux avec n'importe quelle autre étiquette sont considérés comme traitant de transports.

Tâche 2 – Analyse de sentiment : Cette deuxième tâche est orientée analyse de sentiments. L'objectif est de classifier les tweets selon leur polarité : NEUTRE, NEGATIF, POSITIF, MIXPOSNEG

Pré-traitement : Chaque tweet est divisé en mots en utilisant l'espace comme séparateur. Ensuite chaque mot est transformé en liste de caractères. Tous les caractères sont utilisés, même ceux n'apparaissant qu'une seule fois dans le corpus d'entraînement. Pour nous auto-évaluer, les données d'entraînement sont séparées en cinq ensembles égaux pour effectuer de la validation croisée. Enfin, pour chaque runs d'évaluation, nous en utilisons quatre (80% des données) pour l'entraînement, et un –divisé en deux– pour la validation (10%) et l'évaluation (10%).

3.2 Résultats

Le premier tableau compare nos résultat en auto-évaluation avec nos résultat sur le corpus d'évaluation non annoté. On peut voir que nos résultats sont cohérents, notre modèle n'a a priori pas sur-appris sur le corpus d'entraînement. Lorsque l'on s'intéresse au classement. On peut voir que sur la tâche 1 notre modèle est compétitif puisqu'un de nos runs arrive $2^{ème}$ en terme de performance. Sur la tâche 2 en revanche, notre modèle est bien en dessous des autres puisque nous nous classons $6^{ème}$ au mieux [1].

	Partie 1	Partie 2	Partie 3	Partie 4	Partie 5	Moyenne	Evaluation
Tâche 1	83.67	83.88	83.75	85.40	85.39	84.42	83.048
Tâche 2	65.83	66.07	64.65	67.47	67.41	66.28	65.556

TABLE 1 – Précision de classification de notre modèle. Les valeurs présentés à gauche sont celles obtenues en auto-évaluation par validation croisées sur cinq ensembles. A droite, les valeurs présentés sont les résultats obtenus sur les données tenues secrètes

4 Discussion

Ici, pour répondre aux tâches 1 et 2 de l'atelier DEFT2018, nous avons proposé un modèle de représentation hiérarchique des tweets, dérivé des travaux de (Yang *et al.*, 2016). L'originalité de notre modèle est de prendre en compte les diverses spécificités qui font que les tweets sont souvent compliqués à traiter. Ici, nous présentons un modèle avec des pré-traitement minimes et ayant des résultats compétitifs.

1. Après vérification, nous avions entraîné notre modèle à classifier les tweets en INCONNU en plus des sentiments (une classe en plus), de ce fait nous affichons une performance moindre en évaluation. S'il on soustrait les 190 tweets classés comme INCONNU, la performance ce même modèle est $\tilde{6}8\%$ de précision

Equipe	Tâche 1	Rang	Tâche 2	Rang
Lip6 (Nous)	83.048	2	65.824	6
EDF	82.293	5	67.013	4
CLaC	77.955	10	33.494	10
Tweetaneuse	**83.124**	**1**	67.699	3
IRIT	82.433	4	69.906	2
IRISA	82.702	3	**70.258**	**1**
ELOQUANT	81.408	6	66.684	5
EPITA	80.502	8	64.172	7
UTTLM2S	79.580	9	63.004	8
SYLLABS	80.604	7	–	12
ADVteam	70.509	11	29.778	11
LISlab	–	12	47.628	9

TABLE 2 – Performance en précision et rang de l'ensemble des équipes sur les tâches 1 et 2. Les meilleures performances sont en gras.

Pour aller plus loin dans la prise en compte des particularités afférentes aux tweets, les mentions et les hashtags pourraient également avoir des représentations spécifiques, non-liés à celles des lettres. En effet, l'utilisation de modèles hybrides utilisant à la fois les mots et les lettres permettent parfois des gains de performances.

Remerciements

Ce travail a été réalisé en partie avec le soutien du FUI-BIND

Références

CHUNG J., GÜLÇEHRE Ç., CHO K. & BENGIO Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, **abs/1412.3555**.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, **abs/1310.4546**.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L. & TORRES-MORENO J.-M. (2018). Deft2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en île de france. In *Actes de DEFT*, Rennes, France.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543.

YANG Z., YANG D., DYER C., HE X., SMOLA A. J. & HOVY E. H. (2016). Hierarchical attention networks for document classification. In *HLT-NAACL*.

Notre tweet première fois au DEFT-2018 : systèmes de détection de polarité et de transports

David Graceffa Armelle Ramond Emmanuelle Dusserre
Ruslan Kalitvianski Mathieu Ruhlmann Muntsa Padró
Eloquant, 5 allée de Palestine, 38610 Gières, France
{prenom.nom}@eloquant.com

RESUME

Cet article décrit les systèmes de l'équipe Eloquant pour la catégorisation de tweets en français dans les tâches 1 (détection de la thématique *transports en commun*) et 2 (détection de la polarité globale) du DEFT 2018. Nos systèmes reposent sur un enrichissement sémantique, l'apprentissage automatique et, pour la tâche 1 une approche symbolique. Nous avons effectué deux *runs* pour chacune des tâches. Nos meilleures F-mesures (0.897 pour la tâche 1 et 0.800 pour la tâche 2) sont au-dessus de la moyenne globale pour chaque tâche, et nous placent dans les 30% supérieurs de tous les *runs* pour la tâche 2.

ABSTRACT

Systems for detecting polarity and public transport discussions in French tweets
This paper presents Eloquant's team's systems for automatic classification of French tweets in tasks 1 (detection of discussions about public transport) and 2 (detection of the overall polarity) of DEFT-2018. Our systems are based on semantic enrichment, machine learning and, for task 1, a symbolic approach. We performed two runs for each task. Our best F-measures (0.897 for task 1 and 0.800 for task 2) are above the overall average for each task and place us in the top 30% of all runs for task 2.

MOTS-CLES : Tweets, fouille d'opinions, transports, classification automatique
KEYWORDS : Tweets, sentiment analysis, transport, automatic classification

1 Introduction

Cet article décrit les méthodes et les résultats obtenus par l'équipe sémantique d'Eloquant aux tâches 1 et 2 du *Défi Fouille de Textes 2018* (Paroubek et al., 2018), qui concerne l'analyse des thématiques et des opinions exprimées dans un corpus de 68 916 tweets annotés manuellement[1] dans le cadre du projet REQUEST[2].

La **tâche 1** consiste à déterminer automatiquement si un tweet concerne ou non les transports en commun. Il s'agit donc de classification binaire. La **tâche 2** est également une tâche de classification

[1] https://ocsync.limsi.fr/index.php/s/Mbm4Hl5YnALJRKx

[2] Programme d'Investissement d'Avenir, appel Cloud computing & Big Data, convention 018062-25005

et consiste à déterminer, pour les tweets qui concernent les transports en commun (51% du total), leur polarité globale parmi quatre catégories : *positif*, *négatif*, *neutre* et *mixposneg* (à la fois positif et négatif). La partie du corpus d'entrainement annotée par polarité contient environ 37% de tweets négatifs, 36% de tweets neutres, 21% de tweets positifs et 7% de tweets à polarité mixte.

Les outils d'analyse sémantique d'Eloquant ont été développés dans le but de traiter des *verbatim* issus du domaine de la relation client (SMS ou formulaires web). Il n'était donc *a priori* pas trivial pour nous de réaliser les deux tâches (1 et 2) avec nos méthodes symboliques existantes car elles sont composées de règles très spécifiques au style de notre secteur. En effet, l'attitude langagière adoptée dans le cadre de la relation client et celle que l'on trouve sur Twitter sont extrêmement différentes. C'est pourquoi, nous avons suivi deux voies : développer pour l'occasion une méthode symbolique (tâche 1, *run* 1), et proposer des méthodes entièrement statistiques (tâche 1 *run* 2, et tâche 2 *run* 1 et 2). Ces deux approches reposent sur un enrichissement sémantique, nous en parlerons plus précisément dans la suite de cet article.

Dans le reste de l'article nous décrivons les étapes (communes pour toutes les tâches et spécifiques pour chaque tâche) de traitement du corpus, de création de classifieurs, leurs évaluations et la mise en perspective des résultats par rapport à la qualité des données et les limites théoriques des approches utilisées.

2 Chaîne de traitement des données

Qu'il s'agisse de la phase d'apprentissage d'une méthode statistique, de la phase de test ou encore d'une méthode symbolique, nous avons procédé à un travail de représentation des documents, qui se traduit par un pré-traitement, une analyse morphosyntaxique et enfin un enrichissement sémantique.

2.1 Pré-traitement et analyse morphosyntaxique

Afin de préparer notre corpus aux différentes méthodes testées, nous avons effectué une suite de normalisations. En effet, les données issues de Twitter étant particulièrement bruitées il nous est apparu primordial d'en homogénéiser le contenu, en plus de nos traitements habituels.
Nous avons appliqué nos scripts génériques de correction d'encodage des caractères accentués pour les ramener à de l'UTF-8. Ainsi, la séquence "ã©" a par exemple été corrigée en "é" ou encore """ par le caractère ". Les chaînes "[ASCII012CTRLC]" et "[ASCII015CTRLC]" rencontrées dans certains tweets, et qui correspondent à des passages à la ligne, ont été remplacées par un espace lorsqu'ils étaient précédés par un caractère de ponctuation, sinon par un point.

Nous avons également normalisé les émojis en ajoutant un espace lorsqu'apparaissaient plusieurs emojis collés, afin qu'ils puissent être reconnus comme des *tokens*[3] distincts, par exemple ☺☺☺ devient ☺ ☺ ☺. Cela est nécessaire pour l'enrichissement sémantique qui intervient par la suite.

[3] Dans cet article nous préférons cet anglicisme au terme français « item » moins courant, ou « forme », plus ambigu.

Ensuite, nous avons procédé à une normalisation orthographique :

- Suppression des espaces superflus.
- Reponctuation par un point des phrases non ponctuées.
- Suppression des URL.
- Correction orthographique à l'aide de ressources constituées habituellement pour notre cœur de métier, la relation client.
- Suppression de lettres dupliquées dans certaines formes telles que "xptdrrr", "mdrrr", "ptdrrr", "loool". (Ces formes peuvent être des marqueurs importants pour la détection de polarité.)

Enfin, nous avons appliqué les analyses et traitements classiques : segmentation en phrases, tokenisation, lemmatisation, attribution des étiquettes morphosyntaxiques[4], dépendances syntaxiques.

2.2 Enrichissement sémantique

La littérature (Abdaoui et al., 2015, Chen et al., 2011, Vernier et al., 2009) indique que l'enrichissement sémantique peut améliorer les résultats dans des tâches de classification. Nous avons mis cela en œuvre grâce à des *gazetteers* : il s'agit de listes de lemmes où chaque entrée est associée à un *tag* sémantique. Chaque *gazetteer* a été construit et appliqué en fonction d'un type de *PoS* afin de réduire l'ambiguïté : lorsque le système rencontre dans un document un *token* présent dans un *gazetteer* avec la *PoS* correspondante, il attribue à ce *token* le *tag* sémantique associé dans le *gazetteer*. L'annotation sémantique qui en découle peut ensuite être exploitée comme trait dans les règles symboliques ou dans l'apprentissage automatique.

Pour la tâche 1, nous considérons qu'une annotation du lexique relevant de la thématique des transports en commun est pertinente. Nous avons réalisé une extraction semi-automatique des lemmes spécifiques à ce domaine, en plusieurs étapes : extraction des concepts selon la méthode proposée par (Sclano et Velardi, 2007), calcul pour chaque concept de deux fréquences relatives (dans le corpus *transport* et dans le corpus non *transport*), filtrage sur les concepts ayant une fréquence relative supérieure à 0,005 dans l'un ou l'autre des corpus, tri des concepts par ordre décroissant de la différence entre les deux fréquences. La liste ainsi obtenue a ensuite été revue et complétée manuellement afin de créer les *gazetteers* en attribuant à chaque entrée le *tag* sémantique TRANSPORT.

Pour la tâche 2, nous avons également relevé du lexique porteur de polarité (positive ou négative), de façon manuelle cette fois en lisant un certain nombre de tweets. En effet, notre méthode d'extraction semi-automatique décrite plus haut n'est pas adaptée à des notions subjectives comme la polarité. A l'issue de ce travail, nous disposons de *gazetteers* dans lesquels les lemmes relevés sont associés aux *tags* POS[5] ou NEG selon leur sémantique. De plus, l'exercice est un peu plus complexe car il faut ensuite prendre en compte une éventuelle négation (par exemple "Je ne suis pas content"), nous avons donc ajouté une couche de règles symboliques pour traiter les principaux cas rencontrés. Ainsi, l'annotation POS attribuée par un *gazetteer* au *token* "content" va être transformée en NEG par l'application d'une règle de détection de la négation.

[4] Par souci de brièveté nous utiliserons l'abréviation anglaise "*PoS*" par la suite.

[5] POS pour positif, à ne pas confondre avec *PoS*.

3 Création des classifieurs automatiques

3.1 Principes et choix généraux pour l'approche statistique

Afin d'exploiter au mieux la puissance de notre algorithme d'apprentissage automatique et d'obtenir les meilleurs résultats, nous avons eu recours à la validation croisée (80% vs 20%) pour tester plusieurs configurations. Pour des raisons de vitesse d'entrainement, seuls les 20000 premiers tweets ont été utilisés pour nos tests. L'algorithme d'apprentissage automatique supervisé retenu est *liblinear*[6] (Fan et al., 2008) avec les paramètres par défaut. Nous avons bien évidemment veillé à respecter la proportion de chaque catégorie pour la création de nos sous-corpus.

Afin d'obtenir un résultat optimal, nous avons testé plusieurs configurations grâce à la combinaison de différents traits. Nous avons ainsi utilisé : les unigrammes (lemmes, *tokens*), les *PoS*, les dépendances syntaxiques, les groupes nominaux d'une longueur maximale de 4, comme par exemple : "arrêt de bus", "station de métro", etc. et enfin les traits sémantiques qui reposent sur l'enrichissement sémantique.

Nos expérimentations ont montré que nous obtenons les meilleurs résultats en combinant ces différents traits : dépendances syntaxiques, enrichissement sémantique et groupes nominaux. Au regard des résultats, nous avons préféré écarter l'utilisation des *PoS* qui les faisaient décroitre. Nous expliquons ce phénomène en partie par le langage utilisé sur Twitter, qui est non standard, ce qui a tendance à fausser les analyses morphosyntaxiques des outils que nous utilisons.

3.2 Méthodes pour la tâche 1

Pour rappel, la tâche 1 avait pour objectif de distinguer les tweets évoquant les transports en commun des autres tweets. Nous avons mis en œuvre deux méthodes : une symbolique et une statistique. La symbolique se concentre sur l'enrichissement sémantique réalisé (en résumé, la présence de lexique spécifique à la thématique). En comparant les deux méthodes, nous cherchons à savoir si, dans le cas de tweets qui présentent par nature peu de variations, la puissance d'un algorithme d'apprentissage automatique qui prend en compte plusieurs traits, présente un réel intérêt.

3.2.1 Approche symbolique (run 1) : lean and mean

Notre piste de départ était de réaliser une méthode symbolique se basant sur le lexique spécifique au domaine des transports en commun. Nous avons rapidement réalisé que la manifestation de cette thématique dans les tweets est portée par le lexique et que la syntaxe notamment n'apporte pas d'information supplémentaire pour sa détection. Par exemple, dans la phrase "Je suis dans le bus", le *token* "bus" à lui seul permet de déterminer qu'il s'agit d'un tweet évoquant les transports. C'est généralement le cas pour la détection de classes thématiques. Cela est bien différent dans le cas de la tâche 2 comme nous l'avons déjà évoqué.

[6] https://www.csie.ntu.edu.tw/~cjlin/liblinear/

Notre méthode symbolique est donc "*lean and mean*" (ou "bête et méchant") puisqu'elle consiste tout simplement à classer dans la catégorie *transport* tous les tweets comportant un *token* annoté TRANSPORT (cf. 2.2).

3.2.2 Approche statistique (run 2)

Nous avons réalisé un *run* en utilisant les paramètres préalablement définis (cf. 3.1), en particulier l'enrichissement sémantique TRANSPORT décrits en section 2.2. Cette approche nous a paru envisageable car nous avions un corpus annoté conséquent, ce qui représente un socle solide pour un algorithme d'apprentissage automatique.

3.3 Modèles pour la tâche 2

La tâche 2 proposait de détecter la polarité globale d'un tweet évoquant les transports en commun et de le classer parmi les catégories suivantes :

- *positif* : tweet à polarité uniquement positive (21% dans le corpus d'entrainement).
- *négatif* : tweet à polarité uniquement négative (37%).
- *neutre* : tweet sans opinions (36%).
- *mixposneg* : tweet contenant de la polarité positive et négative (7%).

Cette tâche est plus complexe que la première. En effet, il nous était difficile d'élaborer un système symbolique *lean and mean* en nous basant uniquement sur le lexique du corpus comme nous l'avons fait pour la tâche 1. D'autre part, comme nous l'avons expliqué en introduction, les méthodes symboliques d'analyse des sentiments dont nous disposons déjà, ont été développées pour le domaine de la relation client et ne donnent pas de bons résultats sur les tweets.

Nous avons alors décidé d'utiliser uniquement des méthodes statistiques pour cette tâche. Néanmoins, une des limites de cette méthode est la qualité du corpus utilisé pour l'entrainement des modèles. Dans notre cas, le corpus était quelque peu bruité. De même, certaines annotations étaient discutables. Les performances de l'algorithme d'apprentissage automatique ont pu donc être impactées de façon négative par ces différents facteurs. Nous en parlerons plus précisément dans la suite de l'article.
Nous avons alors réalisé une série de tests pour comparer différentes configurations. Nous avons opté pour la construction de deux modèles : pour chacun nous avons le même paramétrage, mais le corpus utilisé lors de l'apprentissage diffère.

3.3.1 Modèle avec la catégorie mixposneg

Le premier modèle a été construit avec l'intégralité du corpus annoté fourni, avec le paramétrage que nous avons décrit plus haut. Pour rappel, nous avons utilisé pour la création de ce modèle l'enrichissement sémantique grâce au lexique polarisé que nous avons extrait du corpus, ainsi qu'un lexique d'émojis polarisés (Novak et al., 2015). Ainsi, nous avons donné une polarité à la quasi majorité des émojis existants.

Nous avons remarqué, à la vue de nos résultats, que les tweets annotés en *mixposneg* étaient parfois très discutables. Il s'agit de plus de la catégorie la moins représentée (7%) et donc de celle pour laquelle nous avons le moins de données pour l'apprentissage du modèle. Lorsque nous entraînions notre modèle avec cette catégorie, nos résultats de précision, rappel et F-mesure tendaient à chuter.

Catégorie de référence	Nb d'occurrences	TP[7]	FP[8]	Précision
mixposneg	461	152	963	14%

TABLE 1 : précision de la catégorie *mixposneg* lors de la validation croisée

C'est pourquoi nous avons décidé d'entrainer un modèle sur un corpus sans la catégorie *mixposneg*. Notre hypothèse était que cette catégorie introduit trop de confusion et qu'il valait donc mieux ne pas l'utiliser.

3.3.2 Modèle sans la catégorie mixposneg

La validation croisée appliquée à un corpus d'entrainement privé de la catégorie *mixposneg* nous a permis d'obtenir de meilleurs résultats (F-mesure, Rappel, Précision). Ces résultats sont logiques, puisque la précision du repérage de la catégorie *mixposneg* est extrêmement basse (cf. table 1) et a un impact négatif sur nos résultats totaux.

Nous avons testé nos deux modèles (avec ou sans *mixposneg* dans le corpus d'apprentissage) sur une partie du corpus d'entrainement réservé à être utilisé comme corpus de test. Ce dernier était alors composé de tous les labels de référence, comprenant également la catégorie *mixposneg*. Nous avons alors obtenu les résultats ci-contre[9].

Configuration des modèles	Précision	Rappel	F-Mesure
Corpus d'entrainement sans *mixposneg*	0,656	0,656	0,656
Corpus d'entrainement avec *mixposneg*	0,643	0,643	0,643

TABLE 2 : comparatif de résultats entre les deux modèles appliqués au même fichier de test (avec *mixposneg*)

Ces résultats se confirment lors de l'évaluation finale (cf Table 3), notre modèle ne comprenant pas de *mixposneg* (*run* 1) étant plus performant que le modèle entrainé avec (*run* 2).

4 Evaluation

Cette section décrit les performances de toutes les équipes aux tâches 1 et 2. Onze équipes ont soumis un total de 38 *runs* à la tâche 1, et 39 *runs* à la tâche 2.

[7] TP : *true positive* (vrai positif)

[8] FP : *false positive* (faux positif)

[9] Les résultats obtenus sont issus de notre propre formule qui diffère de celle utilisée pour le DEFT.

4.1 Evaluation pour la tâche 1

Le tableau ci-dessous récapitule les performances des meilleurs *runs* pour chaque équipe à la tâche 1. Pour notre équipe, nous donnons les chiffres pour nos deux *runs*. Les valeurs marquées avec * ont une différence statistiquement significative[10] avec notre meilleur *run*, avec une valeur-p de 0,01. Les valeurs marquées avec ** sont statistiquement différentes avec une valeur-p de 0,05.

Rang[11]	**N° équipe_n° tâche**	***Run***	**Précision**	**Rappel**	**F1**
1	3_T1	2	0,83124	1	0,90785*
2	7_T1	5	0,83048	1	0,90739*
6	6_T1	4	0,82702	1	0,90532**
11	5_T1	1	0,82433	1	0,90371
12	1_T1	1	0,82293	1	0,90286
18	**8_ELOQUANT_T1**	**1**	**0,81408**	**0,99984**	**0,89745**
19	11_T1	3	0,80604	1	0,8926
21	9_T1	3	0,80502	1	0,89198
MOYENNE			*0,80293*	*0,99997*	*0,89029*
29	10_T1	3	0,79580	1	0,88629*
30	**8_ELOQUANT_T1**	**2**	**0,79360**	**0,99984**	**0,88486***
33	2_T1	2	0,77955	1	0,87612*
37	14_T1	1	0,70509	1	0,82704*

TABLE 3 : performances des équipes à la tâche 1 du DEFT 2018.

La moyenne des F-mesures des 38 *runs* est 0,8882. La plupart des équipes obtient des scores très comparables dans cette tâche (c'est moins le cas pour la tâche 2).

Notre meilleur *run*, correspondant à l'approche symbolique, est au-dessus de la moyenne, avec un F-score de 0,89745, et l'écart entre cette performance et le champion est faible (0,014 point de F-mesure). L'écart entre nos deux *runs* est statistiquement significatif, et d'environ 0,013 points de F-mesure.

4.2 Evaluation pour la tâche 2

Le tableau ci-dessous récapitule les performances des meilleurs *runs* à la tâche 2. Pour notre équipe, nous donnons les chiffres pour nos deux *runs*. Nous avons omis le *run* d'une équipe qui présentait une anomalie (valeurs numériques `NaN`). Les résultats des tests de significativité sont représentés de la même façon que pour la tâche 1.

[10] Selon le test de Ficher

[11] Parmi tous les *runs*

Rang[12]	N° equipe_n° tâche	*Run*	Micro-précision	Micro-rappel	Micro-F1
1	5_T2	5	0,69906	1	0,82288*
4	6_T2	1	0,70258	0,96497	0,81313**
6	3_T2	2	0,67699	1	0,80738
9	1_T2	3	0,67013	1	0,80249
11	**8_ELOQUANT_T2**	**1**	**0,66684**	**1**	**0,80012**
13	**8_ELOQUANT_T2**	**2**	**0,66151**	**1**	**0,79627**
15	9_T2	4	0,64172	1	0,78176*
17	10_T2	1	0,63004	1	0,77304*
19	7_T2	5	0,65824	0,93075	0,77113*
MOYENNE			*0,59364*	*0,95199*	*0,72538**
31	15_T2	3	0,47628	1	0,64524*
37	2_T2	1	0,34255	1	0,5103*
39	14_T2	1	0,29778	0,52821	0,38085*

TABLE 4 : performances des équipes à la tâche 2 de DEFT 2018

La moyenne des F-mesures des 39 *runs* est d'environ 0,73, trois quarts des *runs* ayant une F-mesure supérieure à 0,7. Notre meilleur *run*, correspondant au classifieur entraîné en l'absence de tweets annotés comme *mixposneg*, produit une F-mesure de 0,80. Cela place notre système dans les meilleurs 30% de tous les *runs* pour la tâche 2. L'écart entre ce *run* et le *run* champion est d'environ 0,023 points de F1, et c'est le seul système avec lequel nous observons une différence statistiquement significative (valeur-p à 0.01). L'écart entre nos deux *runs* est d'environ 0,004 points, cette différence n'étant pas significative.

5 Discussion

5.1 Choix et intérêts des méthodes

Pour la tâche 1, nous observons que seulement les deux premiers systèmes ont une performance statistiquement supérieure à celle de notre meilleur *run* (valeur-p = 0.01). En fait, pour cette tâche, la plupart des systèmes ont des résultats comparables et assez élevés. Nous pensons que cela est dû au fait que la tâche est relativement simple, et que, comme nous l'avons vu avec le système symbolique, la distinction peut être fortement basée sur le lexique.

L'approche symbolique a une performance légèrement meilleure que celle de l'approche par apprentissage automatique. Au cours des évaluations que nous avons menées lors des développements, nous avons observé quelques différences intéressantes entre ces deux méthodes. Nos résultats par validation croisée sur le corpus d'entrainement ont notamment présenté un rappel pour la catégorie *transport* plus élevé avec la méthode symbolique. Celle-ci étant basée sur une sélection du lexique

[12] Parmi tous les *runs*

spécifique aux transports en commun, elle permet de bien détecter les tweets de la thématique et de ne pas en laisser de côté. La nature de la tâche 1, qui est nous semble-t-il basique, se prête bien à ce type de méthode *lean and mean* telle que décrite en 3.2.1.

A l'issue de la période de test, nous avons également cherché à comprendre la différence entre nos deux méthodes pour la tâche 1 en observant les tweets pour lesquels les deux *runs* produisaient des sorties différentes. Les cas où la méthode statistique attribue à tort la catégorie *transport*, concernent des tweets qui abordent des sujets proches des transports en commun mais sans les évoquer directement (par exemple : travaux, circulation, horaires, voyages). Ce phénomène est difficile à corriger. A l'inverse, lorsque l'approche statistique attribue à raison la catégorie *transport*, et non l'approche symbolique, cela est dû à sa capacité à prendre en compte des traits variés. Par exemple, lorsqu'une entité nommée était mentionnée par une forme non identifiée par notre lexique : pour le RER E, nous avions listé "RER", "RER_E", "RERE_RATP" mais pas "RERE_SNCF". Un travail de normalisation pourrait corriger cela dans la méthode symbolique. Le fait d'observer les deux méthodes, permet de trouver des pistes pour améliorer les deux. Une poursuite intéressante de ce travail serait de développer une méthode hybride combinant l'intérêt de chacune d'elles, et corrigeant mutuellement leurs faiblesses.

Pour la tâche 2, nos deux *runs* sont très proches et, en prenant en compte les tests de significativité, très bien situés dans le classement global. Même si la différence entre nos deux *runs* n'est pas statistiquement significative, il est intéressant de noter que les meilleurs résultats sont obtenus en ignorant l'annotation *mixposneg* pour entrainer le modèle. Dans ce *run*, le modèle ne va jamais assigner cette catégorie à un tweet, donc ce système a toujours une Précision et un Rappel égale à 0 pour cette catégorie. Néanmoins, les résultats montrent qu'il est préférable d'accepter ces erreurs systématiques que d'essayer de créer un modèle capable de reconnaitre cette catégorie. En effet, la très basse fréquence de cette catégorie dans le corpus et la grande ambiguïté qu'elle présente introduisent trop de bruit pour les systèmes d'apprentissage automatique.

5.2 Qualité du corpus

Les textes issus de médias sociaux, et particulièrement de ceux qui imposent des contraintes sur la longueur des énoncés, présentent des phénomènes morpho-lexicaux et morphosyntaxiques peu standard. De plus, Twitter, étant une plate-forme privilégiée par une population particulière, a développé un sociolecte particulier, comprenant des lexiques, des graphies, et des tournures de phrase propres à ce milieu. La conséquence de cela est l'inadéquation d'outils d'analyse du français plus standard pour le traitement de ces textes.

Le corpus est d'une taille très conséquente, ce qui pousse à penser que, une fois des normalisations effectuées, la quantité des données est suffisante pour que des méthodes classiques d'apprentissage automatique s'approchent de leurs performances maximales théoriques.

Cependant, en parcourant les annotations nous avons constaté des imperfections. Pour de nombreux tweets notre équipe est unanime sur le fait que l'annotation ne correspond pas à la valence émotionnelle ressentie. Le tableau ci-dessous donne quelques exemples ; certains cas de figure sont plus fréquents que d'autres.

Label de référence	Label « ressenti »	Tweet
positif	*négatif*	@nighshxde c' est à cause de ma 4g aussi :(si elle marchait dans le métro Ca m' arrangerai 😭😭😭😭😭😭😭
positif	*négatif*	Avec la putain de rentrée des classes , mon bus est arrivé en retard donc j' ai loupée mon train. Super. Génial.
mixposneg	*négatif*	Panne du ter17758 arrêté au milieu des voies. On cuit dedans avec cette chaleur et tout est fermé déjà 1h de retard merci #SNCF @SNCF
mixposneg	*négatif*	@LIGNEL_sncf pas de bus 275 et pas de train , et évidemment pas de remboursement merci @SNCF fdp
négatif	*neutre*	@SNCF Hello , le site de la SNCF étant en maintenance , pouvez-vous m' indiquer sur le train Lyon/Paris n°6616 de dimanche est annulé ? Merci.
neutre	*négatif*	Question aux parents : il fait 300° dans le bus et 2 gamins se sont mis à chanter 🎶Libérée , délivrée🎶. Ai-je le droit de les tuer ? 🤔 Merci !
neutre	*négatif*	Eh le conducteur du bus ! Le bus c' est pas un kart alors calmes toi !
positif	*neutre*	Retour à un trafic régulier sur l' ensemble de la #Ligne8 #RATP. Incident terminé
négatif	*positif ?*	A Paname je suis choquée ya des prises sur les abris de bus 😐

TABLE 5 : Exemples d'annotations polémiques

Le plus gros problème constaté est l'interprétation quasi-littérale des tweets, l'ironie n'étant que rarement remarquée par les annotateurs. Or elle semble très présente dans les tweets fournis. Toutefois, il ne s'agit pas ici de minimiser l'effort et la difficulté de la tâche d'annotation, car pour certains tweets nous-mêmes n'étions d'accord ni entre nous, ni avec le label de référence.

Il y a quelques rares cas de tweets dupliqués et annotés différemment, comme le suivant, annoté comme *neutre* et *négatif* :

```
"@RERE_SNCF 3€ du + sur le navigo , ça permettra d' avancer autrement
qu' au ralenti avant Villiers pour ne pas rater le bus ? #qml #marre".
```

Il est difficile de quantifier l'impact de ce bruit sur les scores, mais il est clair qu'ils ne peuvent qu'en être dégradés.

5.3 Améliorations possibles

Une étape essentielle du prétraitement des textes bruités est la normalisation. Il existe différentes approches, dont un état de l'art peut être trouvé dans (Tarrade, 2017). Ces approches vont de simples substitutions lexicales prédéfinies, à la traduction automatique (Kaufmann et Kalita, 2010).

Actuellement, nous effectuons une normalisation minimale, qui peut être améliorée. Han et Baldwin (2011) proposent de réduire à deux lettres toute suite de plus de deux lettres identiques à l'intérieur d'une forme (ainsi 'coooool' devient 'cool'). Une approche similaire, dans laquelle on réduirait à une lettre permettrait de ramener des formes 'ptdrrrr', 'looool', 'noooooooon' aux plus fréquentes 'ptdr', 'lol', 'non', respectivement. Nous effectuons déjà un traitement *ad hoc* de certains de ces cas, mais n'avons pas encore généralisé cette approche.

Nous pourrions également exploiter les ressources lexicales de normalisation de langage SMS et les outils de normalisation développés sur des corpus proches (Tarrade et Lopez, 2017), comme le corpus 88milSMS (Panckhurst et al., 2014).

Alternativement, une façon non supervisée d'approcher les données serait d'utiliser des vecteurs de mots word2vec (Mikolov et al, 2013) construits sur un grand ensemble de tweets en français.

Enfin, étant donné les désaccords constatés autour des annotations, une ré-annotation collaborative du corpus pourrait être envisagée. Cela permettrait de marquer en même temps les tweets contenant de l'ironie, caractérisant ainsi quantitativement ce phénomène linguistique difficile à identifier.

6 Conclusion

Nous avons présenté les systèmes de l'équipe Eloquant pour la catégorisation de tweets en français dans les tâches 1 (détection de la thématique transports en commun) et 2 (détection de la polarité globale) du DEFT 2018.

En adaptant nos outils existants, reposant sur un enrichissement sémantique, l'apprentissage automatique et, pour la tâche 1, une approche symbolique, nous avons obtenu des F-mesures au-dessus de la moyenne globale pour chaque tâche, nous plaçant dans les meilleurs 30% de tous les *runs* pour la tâche 2. Il est également notable que pour cette tâche, seul le système avec la meilleure performance a une différence statistiquement significative avec celle de notre système. Cela implique que notre système est en réalité comparable aux systèmes classés en deuxième position, ce que nous considérons un très bon résultat.

Nous considérons très satisfaisants les résultats obtenus dans les deux tâches. Les systèmes que nous avons présentés ont été adaptés à partir du système que nous utilisons habituellement pour le domaine de la relation client (Maurel et al., 2008), qui traite un type de langage assez différent de celui de Twitter et des opinions sur les transports en commun. Ainsi, être comparable aux systèmes qui obtiennent les deux ou trois meilleurs résultats nous parait une indication de la maturité de nos systèmes d'analyse sémantique, qui ont pu être adaptés à ce nouveau domaine avec une intervention assez minimale.

Références

ABDAOUI, A., TAPI NZALI, M. D., AZE, J., BRINGAY, S., LAVERGNE, C., MOLLEVI, C., & PONCELET, P. (2015). ADVANSE : Analyse du sentiment, de l'opinion et de l'émotion sur des Tweets Français. *Présenté à 22ème Traitement Automatique des Langues Naturelles*, Caen, France.

CHEN, M., JIN, X., & SHEN, D. (2011). Short Text Classification Improved by Learning Multi-granularity Topics. *In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three* (p. 1776–1781). Barcelona, Catalonia, Spain: AAAI Press.

FAN, R.-E., CHANG K.-W., HSIEH C.-J., WANG X.-R., ET LIN C.-J. (2008). LIBLINEAR : A library for large linear classification. *Journal of machine learning research n° 9*, Aug. 1871-1874.

GUIBON, G., OCHS, M., & BELLOT, P. (2016). From Emojis to Sentiment Analysis. *In WACAI* 2016.

HAN, B., BALDWIN, T. (2011). Lexical Normalisation of Short Text Messages: Makn Sens a #Twitter. *In Proc eedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (p. 368 – 378). Stroudsburg, PA, USA: Association for Computational Linguistics.

KAUFMANN, M., KALITA, J. (2010). Syntactic normalization of twitter messages. *In International conference on natural language processing,* Kharagpur, India.

MAUREL, S., CURTONI, P., & DINI, L. (2008). A hybrid method for sentiment analysis. *In INFORSID*. http://www.ho2s.com/assets/celi-france_english-2.pdf

MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S., DEAN, J. (2013). Distributed representations of words and phrases and their compositionality. *In Advances in neural information processing systems* (pp. 3111-3119).

NOVAK, P. K., SMAILOVIĆ, J., SLUBAN, B., & MOZETIČ, I. (2015). Sentiment of emojis. *PloS one, 10(12), e0144296.*

PANCKHURST, R., DETRIE, C., LOPEZ, C., MOÏSE, C., ROCHE, M., VERINE, B. (2014). Un grand corpus de SMS en français : 88milSMS.

PAROUBEK, P., GROUIN, C., BELLOT, P., VINCENT CLAVEAU, ESHKOL-TARAVELLA, I., FRAISSE, A., JACKIEWICZ, A., KAROUI, J., MONCEAUX, L., TORRES-MORENO, J.-M. (2018). DEFT2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en Île de France. *In Actes de DEFT*. Rennes, France.

SCLANO, F., & VELARDI, P. (2007). TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities. *In Enterprise Interoperability II* (p. 287-290). Springer London.

TARRADE, L., LOPEZ, C. (2017). Corpus de tweets et de SMS annotés pour l'observation de phénomènes linguistiques en français "non standard". *Actes TALN'2017.*

TARRADE, L. (2017). Normalisation des messages issus de la communication électronique médiée. *Mémoire de M2R. Sciences de l'Homme et Société.* 〈dumas-01666146〉

URIELI, A. (2013). Robust French syntax analysis : reconciling statistical methods and linguistic knowledge in the Talismane toolkit. *Thèse de doctorat. Université Toulouse le Mirail-Toulouse II.*

VERNIER, M., MONCEAUX, L., DAILLE, B., & DUBREIL, E. (2009). Catégorisation des évaluations dans un corpus de blogs multi-domaine. *Revue des Nouvelles Technologies de l'Information*, 45--70.

LSE au DEFT 2018 : Classification de tweets basée sur les réseaux de neurones profonds

Antoine Sainson[1] Hugo Linsenmaier[1] Alexandre Majed[1] Xavier Cadet[1]
Abdessalam Bouchekif[2]
Laboratoire Système & Sécurité de l'EPITA (LSE), Paris, France
(1) prénom.nom@lse.epita.fr
(2) prénom.nom@epita.fr

RÉSUMÉ

Dans ce papier, nous décrivons les systèmes développés au LSE pour le DEFT 2018 sur les tâches 1 et 2 qui consistent à classifier des tweets. La première tâche consiste à déterminer si un message concerne les transports ou non. La deuxième, consiste à classifier les tweets selon leur polarité globale. Pour les deux tâches nous avons développé des systèmes basés sur des réseaux de neurones convolutifs (CNN) et récurrents (LSTM, BLSTM et GRU). Chaque mot d'un tweet donné est représenté par un vecteur dense appris à partir des données relativement proches de celles de la compétition. Le score final officiel est de 0.891 pour la tâche 1 et de 0.781 pour la tâche 2.

ABSTRACT

LSE at DEFT 2018 : Sentiment analysis model based on deep learning

In this article we present the contribution of the LSE at DEFT 2018 for task 1 and 2 which consists in classifying tweets. The goal of the first task was to identify if a tweet is related to transportation or not. The purpose of the second task was to identify the feelings associated to tweets. For both tasks we proposed models based on Convolutional (CNN) and Recurrent Neural Networks (LSTM, BLSTM and GRU). Each word in a given tweet is represented by a dense vector learned from data that is relatively similar to the one of the competition. The official score obtained according to the F-mesure is 0.891 for task 1 and 0.781 for task 2.

MOTS-CLÉS : Analyse d'opinions, plongement de mots, réseaux de neurones profonds, classification thématique.

KEYWORDS: Sentiment analysis, word embeddings, deep learning, text classification.

1 Introduction

La classification de textes est une des tâches importantes du traitement automatique du langage naturel (TALN). Elle consiste à attribuer une catégorie à un texte. La classification thématique et l'analyse des sentiments sont les applications les plus répandues auprès des entreprises. Par exemple, les fournisseurs d'actualités comme *Google Actualités* et *Yahoo Actualités* collectent des informations en provenance des sites d'information afin de les regrouper en thèmes. D'autres compagnies utilisent les tweets pour connaître l'opinion des utilisateurs sur un produit ou un service.

La compétition *DEFT 2018* (Paroubek *et al.*, 2018) propose deux tâches de classification de tweets français. La première consiste à classifier ces tweets selon leur thème (*transport* ou *inconnu*) et la deuxième à indiquer la polarité générale des tweets : *positif*, *négatif*, *neutre* ou *mixposneg*.

Avec les récents progrès en apprentissage profond, la performance des systèmes d'analyse de sentiments s'est considérablement améliorée. Par exemple, les auteurs de (Baziotis *et al.*, 2017) utilisent un réseau de neurones de type *BLSTM* (Bidirectional Long Short-Term Memory) avec des mécanismes d'attention tandis que (Deriu *et al.*, 2016) utilise des réseaux neuronaux convolutionnels (CNN). Les deux systèmes ont respectivement obtenu les meilleures performances lors des compétitions *SemEval* 2016 et 2017.

Les réseaux de neurones récurrents sont particulièrement adaptés aux données séquentielles de tailles variables. C'est pour ces raisons que les réseaux de neurones de type LSTM (Hochreiter & Schmidhuber, 1997), *BLSTM* (Schuster & Paliwal, 1997) et *GRU* (Cho *et al.*, 2014) sont les plus utilisés dans le traitement automatique du langage naturel (*e.g.* traduction automatique, analyse des sentiments, chatbot). Les CNNs ont l'avantage de prendre en compte le contexte des données d'entrée, ce qui permet d'avoir de bonnes performances non seulement dans l'analyse d'images, mais aussi dans l'analyse des données textuelle.

Cet article présente notre modèle d'analyse des sentiments basé sur la combinaison de quatre réseaux de neurones profonds. Ces derniers, prennent en entrée la représentation vectorielle des mots (*word embeddings*). Aucun corpus annoté ou lexique externe n'ont été utilisés, les données proviennent uniquement des corpus de test fournis.

L'article est structuré comme suit : dans la section 2, on décrit les prétraitements effectués, la représentation vectorielle des mots ainsi les différents modèles. La section 3 présente les résultats obtenus dans les deux tâches. Enfin, des expériences complémentaires sont décrites dans la section 4.

2 Systèmes Proposés

Dans cette section, nous présentons les trois étapes de notre approche :

1. **Prétraitements :** pour filtrer les imperfections des tweets.
2. **Word embeddings :** pour donner une représentation vectorielle des mots.
3. **Apprentissage des modèles :** pour classifier un tweet selon le thème ou le sentiment exprimé.

2.1 Prétraitements

Pour l'analyse de sentiments, les tweets contiennent plusieurs sources de bruit. L'étape de prétraitement consiste à les préparer pour un traitement automatique efficace.

2.1.1 Lemmatisation

Les mots d'une langue donnée sont accordés en genre, en nombre et en mode (indicatif, impératif...). Le rôle d'un lemmatiseur est de ramener le mot à sa forme canonique (*i.e.* les verbes à l'infinitif et les autres mots au masculin singulier). Ce processus permet de réduire la taille du vocabulaire et d'uniformiser nos tweets. Dans ce travail, nous avons utilisé l'outil *TreeTagger* [1].

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

2.1.2 Normalisation

Cette étape consiste à convertir tous les mots en minuscule, supprimer les informations parasites telles que les URLs, les emails, les dates, les pseudos et les mots non porteurs de sens (*e.g.* le, de, ce, *etc.*). Néanmoins, les mots de négation comme *ne* et *pas* ont été retenus, car la négation dans une phrase peut inverser le sentiment du tweet (Das & Chen, 2001). Les smileys donnent aussi des informations supplémentaires sur les émotions de l'internaute. De ce fait, nous regroupons les smileys ayant une signification commune (tristesse, joie, colère, *etc.*).

2.2 Représentation vectorielle des mots

Les mots de chaque tweet sont représentés par des vecteurs de 104 et 102 dimensions respectivement pour les tâches 1 et 2. Ils ont été obtenus à partir caractéristiques suivantes :

2.2.1 Word embeddings

Le *word embedding* est une représentation de mots dans un espace à n dimensions apprise à partir de réseaux de neurones. Chaque mot est représenté par un vecteur de nombres réels capturant la sémantique des mots. Deux mots sont considérés comme similaires si leurs représentations vectorielles sont proches dans le même espace continu. Notre choix fut de construire notre propre représentation vectorielle, à partir de tweets récoltés sur internet, pour disposer d'une représentation vectorielle plus robuste et adaptée au problème que certains modèles trouvables sur le net. Nos sources proviennent d'1 million de tweets de différentes catégories relatives aux données de la compétition : $\#RATP(169k)$, $\#SNCF(453k)$, $\#IleDeFrance(90k)$, ainsi qu'une collection de tweets traitant de la politique française [2].

Les représentations vectorielles ont été obtenues en utilisant l'outil *Fasttext* [3], qui considère les mots comme des assemblages de caractères (Bojanowski *et al.*, 2016). Ainsi, les fautes d'orthographe n'influent que peu la différence de représentation des mots dans l'espace vectoriel engendré par le Word2vec. Nous fixons la taille maximale de la fenêtre du contexte à 4 et le nombre d'occurrences minimales à 5. Nous choisissons l'architecture *skip-gram* (Mikolov *et al.*, 2013) qui entraîne un réseau de neurones à prédire le contexte d'un mot donné. Les words embeddings utilisés dans ce travail sont de 100 dimensions.

2.2.2 Valence émotionnelle des mots

Pour la tâche 2, à partir des données d'apprentissage, nous associons à chaque mot un vecteur de valeurs comprises entre 0 et 1 représentant sa probabilité d'apparition dans les catégories suivantes : *positif*, *négatif*, *neutre* ou *mixposneg*.

2.2.3 Probabilité thématique des mots

Pour la tâche 1, à partir des données d'apprentissage, nous associons à chaque mot la probabilité d'apparition dans les deux catégories suivantes : *transport* et *inconnu*.

2. http://ideo2017.ensea.fr/projet-polititweets/
3. https://fasttext.cc/

2.3 Apprentissage des Modèles

Pour construire nos modèles, nous avons utilisé deux familles de réseaux de neurones profonds : convolutifs (CNN) et récurrents ($LSTM$, $BLSTM$ et GRU).

2.3.1 Réseaux de neurones récurrents

Les réseaux de neurones classiques supposent que toutes les entrées sont indépendantes les unes des autres, *c'est-à-dire* qu'ils ne prennent pas en compte l'ordre des entrées (dans notre cas, les mots). En revanche, les réseaux de neurones récurrents (RNN) traitent les données au fur et à mesure, tout en respectant l'ordre des entrées. Les RNNs sont principalement conçus pour modéliser des séquences et sont capables de mémoriser des informations passées.

Réseaux de neurones récurrents simples
Un RNN simple à l'instant t observe l'entrée $x^{(t)}$ (un mot dans un tweet) et met à jour la sortie de la couche cachée $h^{(t)}$ en prenant en compte le vecteur $h^{(t-1)}$ calculé à l'étape précédente. Du fait que chaque neurone conserve l'information des étapes précédentes, il est qualifié de *cellule de mémoire*. La sortie de la couche à l'étape temporelle t est donnée dans l'équation 1

$$h^{(t)} = f(W_{xh}\, x^{(t)} + W_{hh}\, h^{(t-1)} + b_h) \tag{1}$$

- f est la fonction d'activation.
- $x^{(t)} \in \mathbb{R}^{\mathrm{d}}$ est le vecteur du mot à l'instant t
- $W_{xh} \in \mathbb{R}^{\mathrm{d,n_h}}$ est la matrice des poids entre l'entrée $x^{(t)}$ et la couche cachée h, sachant que n_h est le nombre de neurones dans la couche cachée.
- $W_{hy} \in \mathbb{R}^{\mathrm{n_h,n_o}}$ est la matrice entre la couche cachée et la sortie, sachant que n_o est le nombre de neurones dans la couche de sortie.
- $W_{hh} \in \mathbb{R}^{\mathrm{n_h,n_h}}$ est la matrice des poids entre la sortie des couches cachées courante et précédente.

Les RNNs simples souffrent de la disparition/explosion des gradients. En effet, plus le modèle est profond, plus les valeurs des gradients propagées vers les couches basses sont affaiblies (*i.e.* les poids deviennent de plus en plus petits). Par conséquent, la mise à jour par descente de gradient a un impact très limité et la convergence vers l'optimum global n'est pas garantie. L'opposé de la disparition des gradients peut se produire, c'est-à-dire que la mise à jour des poids entraîne des modifications trop importantes, ce qui fait diverger le réseau. L'utilisation des fonctions d'activation non saturantes comme *RELU* (rectified linear unit) ou sa variante *ELU* (Exponetial Linear Unit) pourront alléger ce phénomène.
Le RNN simple souffre d'un autre problème, celui de la disparation des premières entrées en mémoire, qui vient principalement de la taille de la séquence. Prenons le tweet "*j'ai adoré les trains de la ligne L, il manque un peu de nettoyage de temps en temps*". Si le RNN perd de l'information sur les premiers mots, le système pourra considérer le message comme ayant une polarité négative.

LSTM, GRU

Les réseaux de type LSTM et GRU sont des variations de RNN simple, capables d'apprendre les dépendances à long terme. Cette capacité vient des cellules de mémoire. Chaque cellule LSTM est liée, non seulement à $h^{(t-1)}$ mais également à un état de la cellule c de l'étape précédente qui joue le rôle de mémoire. $h^{(t)}$ peut être vu comme l'état à court terme et $c^{(t)}$ comme l'état à long terme (Géron, 2017). Une cellule LSTM contient :

- 2 entrées : l'ancien état de la cellule $c^{(t-1)}$ et le vecteur $h^{(t-1)}$
- 4 couches intégralement connectées.
- 3 types de porte : porte d'oubli (f), porte d'entrée (i) et porte de sortie (o). Ces portes utilisent la fonction d'activation *sigmoïde*.
- 2 sorties : $c^{(t)}$ et $h^{(t)}$.

Une cellule LSTM a la capacité d'éliminer, d'ajouter et de stocker des informations dans l'état à long terme $c^{(t)}$. Les équations ci-dessous illustre le calcul de $c^{(t)}$ et $h^{(t)}$

$$
\begin{aligned}
i^{(t)} &= \sigma(W_{xi}\, x^{(t)} + W_{hi}\, h^{(t-1)} + b_i) \\
f^{(t)} &= \sigma(W_{xf}\, x^{(t)} + W_{hf}\, h^{(t-1)} + b_f) \\
o^{(t)} &= \sigma(W_{xo}\, x^{(t)} + W_{ho}\, h^{(t-1)} + b_o) \\
g^{(t)} &= \tanh(W_{xg}\, x^{(t)} + W_{hg}\, h^{(t-1)} + b_g) \\
c^{(t)} &= f^{(t)} \otimes c^{(t-1)} + i^{t} \otimes g^{(t)} \\
h^{t} &= \tanh(c^{(t)}) \otimes o^{(t)}
\end{aligned}
$$

W_{xi}, W_{xf}, W_{xo}, W_{hi}, W_{hf}, W_{ho} et W_{hg} sont des matrices de poids.

b_i, b_f, b_o et b_g sont les biais de chacune des quatre couches.

$\otimes$ est le produit matriciel de Hadamard.

La porte d'oubli permet de filtrer les informations provenant du précédent état à long terme $c^{(t-1)}$. Puisqu'elle utilise la fonction d'activation sigmoïde qui sort des valeurs entre 0 et 1, et en les multipliant par $c^{(t-1)}$, la cellule LSTM filtre les informations de $c^{(t-1)}$. C'est la porte d'entrée $i^{(t)}$ et la sortie de la couche $g^{(t)}$ qui sont responsables de la mise à jour de l'état à long terme. La porte de sortie $o^{(t)}$ est chargée de sélectionner les informations de l'état à long terme à produire à la sortie de la cellule (dans h_t).

La cellule GRU est une version simplifiée de la cellule LSTM (*i.e.* contient moins de paramètres). À la différence des réseaux de type LSTM, dans les réseaux de type GRU :

- On trouvera uniquement deux types de portes : reset gate (r) et update gate (z).
- Les deux vecteurs d'états sont fusionnés en un seul vecteur h_t

L'équation ci-dessous illustre le calcul $h^{(t)}$

$$
\begin{aligned}
z^{(t)} &= \sigma(W_{xz}\, x^{(t)} + W_{hz}\, h^{(t-1)} + b_z) \\
r^{(t)} &= \sigma(W_{xr}\, x^{(t)} + W_{hr}\, h^{(t-1)} + b_r) \\
g^{(t)} &= \sigma(W_{xg}\, x^{(t)} + W_{hg}\, h^{(t-1)} + b_g) \\
h^{(t)} &= z^{(t)} \otimes h^{(t-1)} + (1 - z^{(t)} \otimes g^{(t)})
\end{aligned}
$$

BLSTM
Les réseaux de type BLSTM consistent à exécuter deux LSTM en parallèle : le premier réseau lit la séquence d'entrée de droite à gauche et le second réseau en sens inverse de gauche à droite. Chaque LSTM engendre une représentation cachée : $\vec{h}$ (un vecteur allant de gauche à droite) et $\overleftarrow{h}$ (un vecteur allant de droite à gauche) qui sont ensuite combinés afin de calculer la séquence de sortie. Dans notre problème, saisir le contexte des mots de chaque direction permet de mieux comprendre la sémantique d'un tweet.

2.3.2 Réseau neuronal convolutif

Une architecture de CNN typique consiste en la superposition de plusieurs couches de convolutions, mise en commun (*pooling*) et intégralement connectées. Dans la couche de convolution, chaque neurone est connecté à une région de l'entrée. L'opération de convolution consiste à appliquer une petite fenêtre de poids (également appelé noyau de convolution, filtre ou détecteur de caractéristiques), appliqués à des caractéristiques locales.
Nous avons implémenté avec Keras [4] des CNNs ayant respectivement 1, 2 et 3 couches de convolution pour lesquels nous avons fait varier divers paramètres.
Une autre variante du CNN consiste à appliquer une seule couche de convolution en entrée, en utilisant des noyaux de convolution de différentes tailles dans des réseaux séparés. Les vecteurs de caractéristiques de chaque réseau sont ensuite rassemblés pour n'obtenir qu'un vecteur de caractéristiques. Ce vecteur est ensuite donné en entrée à des couches connectées pour classer le tweet. Ce modèle figurait comme une option intéressante pour la classification de phrases (Zhang & Wallace, 2015).

3 Résultats et analyse

Nous avons opté pour l'utilisation de différents types de réseaux profonds : notre première approche fut de maximiser individuellement le score de chacun de ces réseaux et de mieux comprendre les données de test à partir de leurs sorties.

CNN : Il s'agit d'un CNN d'une couche de convolution de 64 neurones et une fenêtre de taille 5. On applique ensuite un max pooling global avant d'insérer une couche de 128 neurones, un *dropout* de 0.1 et la couche de sortie.

4. https://keras.io/

CNN 2 : Le modèle est composé de deux couches de convolutions de 64 filtres et d'une taille de fenêtre de 5. Un *max pooling* est appliqué après chaque convolution. On insert un *dropout* de 0.3 après la première couche de convolution. Enfin, on ajoute une couche dense de 64 neurones suivie par une couche de sortie de taille correspondante au nombre de classes.

CNN 3 : On définit 3 sous modèles avec les paramètres suivants : une couche de convolution de 64 neurones et des fenêtres de tailles 3, 4 puis 5. On y ajoute un *max pooling* global. Ces 3 modèles sont fusionnés grâce à la couche *Merge* de Keras. On ajoute par la suite un *dropout* de 0.5 suivi d'une couche dense de 64 neurones et de la couche de sortie.

BLSTM : La taille du LSTM est de 64 neurones (128 pour le BLSTM). On y ajoute un *dropout* de 0.5 suivi de la couche de sortie.

GRU : Une couche GRU de 64 neurones suivie de la couche de sortie.

LSTM : Une couche LSTM de 64 neurones suivie de la couche de sortie.

Combinaison 1 : Une combinaison de nos meilleurs modèles : les sorties de chaque modèle nourrissent un réseau de neurones de 3 couches cachées de 20, 20 et 10 neurones. Entre les 2 premières couches, on applique un *dropout* de 0.2.

Combinaison 2 : Une moyenne des prédictions de chaque modèle.

3.1 Analyse

3.1.1 Tâche 1

Le tableau 1 présente la répartition des données en fonction des deux catégories étudiées : *transport* et *inconnu*. Le corpus d'apprentissage contient 59990 tweets : 51.6% *transport* et 48.4% *inconnue*. Le corpus de test est composé de 7816 tweets : 50.4% *transport* et 49.6% *inconnue*

	Transport	Inconnu	Total
Dev.	30951	29039	59990
Test	3941	3875	7816

TABLE 1 – Proportion des données de test et d'entraînement de la tâche 1.

Type de réseau	Précision	F1-mesure
CNN	0.799	0.888
CNN 2	0.798	0.888
CNN 3	0.803	0.890
BLSTM	0.810	0.892
GRU	0.800	0.889
LSTM	0.795	0.886
Combinaison 1	0.767	0.868
Combinaison 2	**0.808**	**0.894**

TABLE 2 – Précision et F1-mesures des données de test de la tâche 1

Le tableau 2 présente les résultats obtenus à la tâche 1. En utilisant, les réseaux de neurones convolutifs

on obtient des performances similaires. Avec les réseaux de neurones récurrents la *F-mesure* prend des scores allant de 0.886 à 0.892. Le meilleur système est celui qui combine les 5 modèles : *CNN, CNN2, BLSTM, GRU et LSTMI* avec une F-mesure de 0.984%. Présenter les sorties des 5 modèles à un réseau de neurone dense (combinaison2) dégrade les performances de notre système de classification

La figure 1 met en avant le fait que notre modèle classifie correctement le plus souvent les tweets concernant les transports.

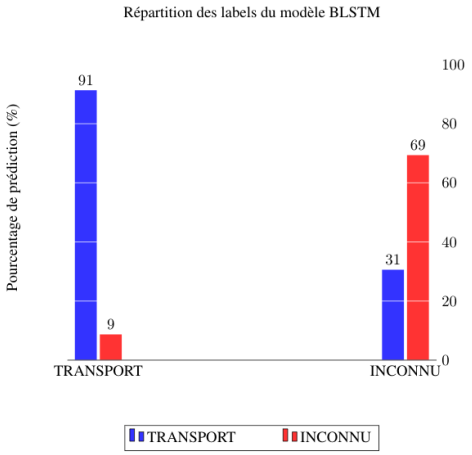


FIGURE 1 – Répartition des prédictions par label de référence du modèle BLSTM de la tâche 1

En effet, notre système détecte facilement les mots comme "bus" ou "métro" comme étant spécifiques aux tweets concernant les transports. Cependant, il arrive parfois que ces mots soient contenus dans des tweets labéllisés *non transport*, ce qui fausse complètement l'analyse du reste du tweet, tant ces mots ont une forte appartenance à la classe *transport*. Les mots les plus susceptibles d'induire en erreur notre système ont été regroupés dans le tableau 3 :

Mots-clés	occurrences
pas	327
métro	158
train	113
sncf	92
rer	88

TABLE 3 – Mots les plus apparents pour les erreurs d'annotations *non transport*

3.1.2 Tâche 2

Le tableau 4 présente la répartition des données en fonction des quatre catégories étudiées *positif*, *négatif*, *neutre* et *mixposneg*. Le corpus d'apprentissage est composé de 35455 tweets (20.7% *positif*, 37.0% *négatif*, 35.6% *neutre* et 6.7% *mixposneg*). Le corpus de test contient 3941 tweets (21.7%

	Positif	Négatif	Neutre	MixPosNeg	Total
Dev.	7324	13105	12609	2417	35455
Test	857	1525	1304	255	3941

TABLE 4 – Proportion des données de test et d'entraînement de la tâche 2

positif, 8.7% *négatif*, 33.1% *neutre*, 6.5% *mixposneg*).
Le tableau 5 présente les résultats obtenus dans la tâche 2. Nous constatons que la combinaison est profitable à notre tâche de classification. Si on prend les performances des systèmes individuellement le GRU s'avère le plus efficace.

Type de réseau	Précision	F1-mesure
CNN	0.605	0.754
CNN 2	0.610	0.757
CNN 3	0.639	0.780
BLSTM	0.623	0.768
GRU	0.641	0.781
LSTM	0.632	0.774
Combinaison 1	0.610	0.757
Combinaison 2	**0.657**	**0.793**

TABLE 5 – Précision et F1-mesures des données de test de la tâche 2

Les différents graphiques de la figure 2 mettent en avant le fait que les systèmes détectent avec une meilleure précision les tweets de polarité négative ou neutre (respectivement 67% et 77%). Les tweets de polarité mixte (*mixposneg*) sont mal interprétés (19% de vrais positifs). Cet écart peut s'expliquer par la faible proportion de tweets de ce type dans les données d'entraînement (6.8%). On remarquera également pour ce type de tweet que le label le plus souvent attribué est *négatif* (42%). Les tweets *positifs* sont souvent confondus comme étant des tweets de type *neutres*.

Ces premières observations faites, on pourra s'intéresser à la comparaison des erreurs des différents systèmes. En effet, certains modèles comme le LSTM détectent mieux les tweets *positifs* (60%), le CNN 1 parvient mieux à identifier les tweets *mixposneg* (23%). Ces différences d'interprétation des modèles pouvant aboutir à une meilleure classification, nous avons conduit d'autres expériences consistant à les combiner. Une moyenne des prédictions nous a permis d'améliorer la précision de 1.6%.

Il faut noter que les données contiennent des erreurs d'annotation, qui peuvent fausser l'évaluation des systèmes de classification. Par exemple, les tweets "Ya intérêt que le bus soit à l'heure parce que en retard le premier jour c'est moyen" et "Si vs avez le bac svp ne pensez pas à vous suicider sur la ligne du RER D merci" sont considérés comme *mixposneg* dans la classification de référence, et *négatifs* par notre système.

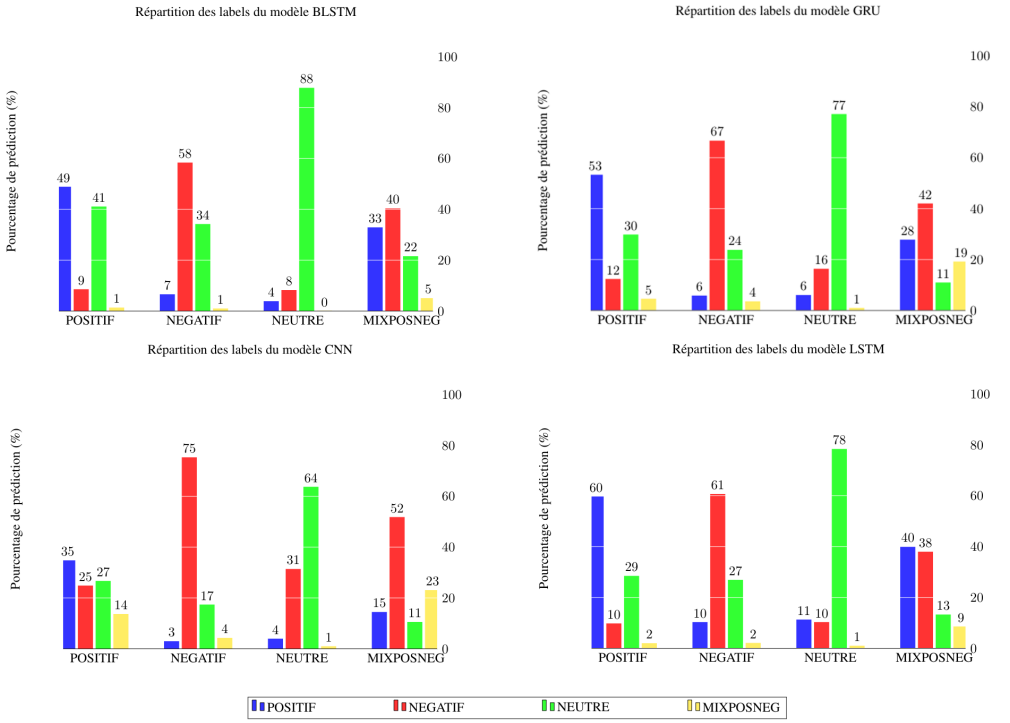


FIGURE 2 – Répartition des prédictions par label de référence des différents modèles de la tâche 2

4 Conclusion

Dans cet article, nous proposons différentes approches basées sur les réseaux de neurones profonds pour classifier les tweets selon leur sujet (*transport* ou *inconnu*) et leur polarité. Pour construire nos plongements de mots, nous avons récolté près d'1 million de tweets traitant de sujets relatifs aux transports. Pour les deux tâches, les résultats obtenus selon l'architecture employée ont des tendances similaires.

Références

BAZIOTIS C., PELEKIS N. & DOULKERIDIS C. (2017). Datastories at semeval-2017 task 6 : Siamese LSTM with attention for humorous text comparison. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada*.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2016). Enriching word vectors with subword information. *CoRR*, **abs/1607.04606**.

CHO K., VAN MERRIENBOER B., GÜLÇEHRE Ç., BAHDANAU D., BOUGARES F., SCHWENK H. & BENGIO Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical

machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,*, p. 1724–1734.

DAS S. & CHEN M. (2001). Yahoo ! for amazon : Extracting market sentiment from stock message boards. In *In Asia Pacific Finance Association Annual Conf. (APFA)*.

DERIU J., GONZENBACH M., UZDILLI F., LUCCHI A., LUCA V. D. & JAGGI M. (2016). Swisscheese at semeval-2016 task 4 : Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, USA*.

GÉRON A. (2017). *Hands on Machine Learning with scikit-learn and Tensorflow*. O'Reilly Media.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, **abs/1310.4546**.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L. & TORRES-MORENO J.-M. (2018). Deft2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en île de france. In *Actes de DEFT*, Rennes, France.

SCHUSTER M. & PALIWAL K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*.

ZHANG Y. & WALLACE B. C. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *CoRR*, **abs/1510.03820**.

Syllabs@DEFT2018 : combinaison de méthodes de classification supervisées

Chloé Monnin[1] Olivier Querné[1] Olivier Hamon[1]
(1) Syllabs, 35-37 rue Chanzy, 75011, France
monnin@syllabs.com, querne@syllabs.com, hamon@syllabs.com

RESUME

Nous présentons la participation de Syllabs à la tâche de classification de tweets dans le domaine du transport lors de DEFT 2018. Pour cette première participation à une campagne DEFT, nous avons choisi de tester plusieurs algorithmes de classification état de l'art. Après une étape de prétraitement commune à l'ensemble des algorithmes, nous effectuons un apprentissage sur le seul contenu des tweets. Les résultats étant somme toute assez proches, nous effectuons un vote majoritaire sur les trois algorithmes ayant obtenus les meilleurs résultats.

ABSTRACT

Syllabs@DEFT2018: Combination of Supervised Classification Methods
This paper describes Syllabs' participation in the task of tweet classification in the transport domain during DEFT 2018. For this first participation in a DEFT campaign, we have chosen to test several state-of-the-art classification algorithms. After a pre-processing step shared by all the algorithms, training is made based only on tweet content. The results being quite close, we realise a majority pool on the three best algorithms.

MOTS-CLES : Classification, SVM, régression logistique
KEYWORDS: Classification, SVM, Logistic Regression

1 Introduction

Les réseaux sociaux prenant une place croissante sur le Web et avec l'avènement de messages courts à des fins de communication, contextualiser l'information est de première importance. Le domaine de la classification vise à organiser, voire hiérarchiser, des *choses* (connaissances, concepts, objets, etc.) selon des classes établies au préalable. Cette classification permet, entre autres utilisations, de filtrer ces *choses* pour focaliser une utilisation, une étude, un traitement, etc.

Dans le cadre de la campagne DEFT 2018 (Paroubek et al., 2018), Syllabs a participé à la première tâche qui a pour objectif de classifier des tweets, selon qu'ils traitent du transport ou non. Dans le cadre de cette tâche un corpus de près de 70 000 tweets annotés nous a été fourni, que nous avons utilisé pour l'entraînement de 5 classifieurs supervisés, et après application de 5 prétraitements différents. De l'ensemble des 25 combinaisons, nous en avons tiré les 4 meilleurs, auquel s'est ajouté un vote des 3 meilleurs, pour soumettre 5 *runs* à la tâche de classification.

Dans un premier temps, nous revenons sur la description de la tâche à laquelle nous avons participé, puis nous décrivons notre méthode de travail, les données et classifieurs utilisées et les résultats obtenus sur un corpus de développement à partir de plusieurs combinaisons de modèles et prétraitements sur les données. Enfin, nous présentons les résultats obtenus sur le corpus de test avant de conclure.

2 Méthode

Nous avons concentré nos travaux sur la première tâche de DEFT, ce qui nous a permis de réaliser plusieurs combinaisons de tests afin de fournir un classifieur ayant les meilleurs résultats sur un corpus de développement.

Outre une sélection des données qui distingue corpus d'entraînement et de développement, notre méthode repose sur des principes simples, à savoir :

1. Prétraitement des données : nous avons testé 5 combinaisons de prétraitements divers.
2. Entraînement de 5 classifieurs supervisés sur le corpus d'entraînement appliqués à chaque prétraitement.
3. Observation des résultats, éventuellement adaptation des étapes 1 et 2.
4. Sélection de 4 *runs* issus des combinaisons prétraitement/modèle ayant obtenues les meilleurs résultats sur le corpus de développement.
5. Ajout du résultat d'un vote majoritaire sur les 3 meilleurs classifieurs en plus des 4 runs sélectionnés à l'étape 4.

Ainsi, 5 *runs* ont été soumis à la première tâche DEFT.

2.1 Sélection des données

Afin d'entraîner les classifieurs et de réaliser nos tests, nous avons séparé le corpus de tweets initial en deux corpus distincts. Le premier, servant de corpus d'entraînement, contient 90 % du corpus initial (soit 62 024 tweets). Le second, utilisé comme corpus de développement, contient les 10 % restant (soit 6 890 tweets).

La Table 1 présente les statistiques d'annotation sur les corpus d'entraînement et de développement, ainsi que le nombre total de tweets (nous avons ajouté les statistiques sur le corpus de test comme comparaison).

Corpus	**#tweets**	**#Transport**	**#Non-transport**
Entraînement	62 024	31 889	30 136
Développement	6 890	3 580	3 311
Test	7 815	-	-

TABLE 1 : prétraitements testés

La répartition des annotations transport et non-transport semble bien répartie que ce soit pour le corpus d'entraînement ou de développement. La taille du corpus de développement est comparable à celui de test.

2.2 Prétraitement

L'analyse de tweets nécessite *a priori* des traitements spécifiques dus à la nature particulière des données. L'information transmise y est réduite à son plus simple concept et brièveté de chaque tweet rend cette information difficile à interpréter. En particulier, le bruit est l'une des plus importantes caractéristiques à filtrer. Dans les tweets, le bruit prédomine généralement sur le contenu pertinent, mais les plus petits extraits d'information peuvent avoir leur importance. Ainsi, la préparation et le prétraitement des données sont des éléments essentiels ayant un impact sur la suite des traitements.

Afin d'étudier cet impact, nous avons mis en œuvre 5 prétraitements différents (*PT1 à 5*). Ceux qui nous fourniront les meilleurs résultats combinés aux classifieurs seront susceptibles d'être conservés :

- **Tokenisation** : le découpage des tweets a été réalisé à l'aide du tokenizer nltk (Loper & Bird, 2002)[1] ;
- **Suppression des URLs** : toutes les chaînes de caractères identifiées comme étant une URL sont supprimées ;
- **Suppression de la casse** : tous les caractères sont convertis en minuscule ;
- **Suppression de la ponctuation** : tous les caractères de ponctuation sont supprimés ;
- **Suppression des mots vides** : les mots vides sont supprimés, à partir d'une liste interne d'environ 300 termes ;
- **Pondération du lexique transport** : la pondération du lexique transport utilise une liste de plus de 1 100 termes et expressions résultant de l'analyse manuelle du corpus d'entraînement. Cette pondération consiste à doubler ces termes et expressions lors de la construction des modèles de classification.

La Table 2 présente les différentes combinaisons de prétraitements sélectionnées[2].

Prétraitements	PT1	PT2	PT3	PT4	PT5
Tokenisation	X	X	X	X	X
Suppression des URLs	X	X	X	X	X
Suppression de la casse		X	X	X	X
Suppression de la ponctuation (hors @ et #)		X	X	X	X
Suppression des mots vides			X		X
Pondération du lexique transport				X	X

TABLE 2 : prétraitements testés

[1] http://www.nltk.org/api/nltk.tokenize.html

[2] Nous n'avons pas jugé utile de combiner l'ensemble des traitements, certains de ces traitements nous semblant indispensables.

2.3 Construction des modèles

Nous avons choisi de tester 5 solutions de classification supervisée, parmi les plus courantes pour ce type de tâche. Pour chacune de ces solutions, à l'exception de l'approche naïve, nous avons fait varier leurs paramètres.

1. **Classification bayésienne naïve (CBN)** : Nous avons utilisé un premier classifieur naïf comme point de comparaison pour évaluer les autres classifieurs. Celui-ci utilise la distribution de la variable de Bernoulli[3].
2. **Machine à vecteurs de support[4] (MVS)** (*Support Vector Machine – SVM*) : Les noyaux RBF et linéaire ont été testés, le second ayant été laissé de côté car moins performant. Nous avons fait varier le paramètre de *C* entre des valeurs de 10^{-1} et 10^{5}.
3. **Régression logistique[5] (RL)** *(Logistic Regression)* : Nous avons joué sur les contraintes du modèle en faisant varier le paramètre de *C* entre des valeurs de 10^{-1} et 30.
4. **Arbre de décision[6] (AD)** (Decision Tree) : Nous avons réalisé des tests utilisant deux stratégies différentes de séparation des nœuds, *random* et *best*.
5. **Descente de gradient stochastique[7] (DGS)** (*Stochastic Gradient Descent – SGD)* : Nous avons fait varier le terme de régularisation avec les paramètres *alpha* (10^{-4} et 10^{-6}) et *penalty* (*l2*, *elasticnet*), ainsi que le nombre d'itération sur l'ensemble d'entraînement (10 à 80).

2.4 Résultats sur les données de développement

L'ensemble des 25 combinaisons (5 prétraitements différents, 5 classifieurs différents) a été évalué sur le corpus de développement en terme de précision (ligne du haut), rappel (ligne du milieu) et f1-mesure (ligne du bas).

Les résultats sont présentés dans la Table 3.

Lorsque l'on observe les résultats de l'évaluation sur notre corpus de développement dans leur ensemble, deux choses en particulier sautent aux yeux. Tout d'abord, les prétraitements n'améliorent pas beaucoup les performances, voire les abaissent. Ensuite, les écarts entre les classifieurs ne sont pas très importants.

Parmi les prétraitements, les suppressions de la casse et de la ponctuation affaiblissent les performances des classifieurs, de même que la pondération à partir d'un lexique transport (ce qui n'est pas le cas pour tous les classifieurs). Au contraire, la suppression des mots vides semble améliorer les performances, même si là aussi ce n'est pas le cas pour tous les classifieurs.

Il faut également noter que le vote majoritaire, de manière surprenante et mis à part pour quelques cas, n'apporte pas d'amélioration significative des résultats.

3 http://scikit-learn.org/stable/modules/naive_bayes.html#bernoulli-naive-bayes
4 http://scikit-learn.org/stable/modules/svm.html#classification
5 http://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
6 http://scikit-learn.org/stable/modules/tree.html#classification
7 http://scikit-learn.org/stable/modules/sgd.html

Modèle	PT1	PT2	PT3	PT4	PT5
Classification bayésienne naïve (CBN)	0,836 0,817 0,819	0,836 0,817 0,819	0,841 0,819 0,821	0,834 0,812 0,815	0,833 0,811 0,814
Machine à vecteurs de support (MVS)	0,864 0,837 0,840	0,862 0,834 0,837	0,845 0,824 0,827	0,859 0,825 0,829	0,847 0,822 0,825
Régression logistique (RL)	0,863 0,835 0,838	0,861 0,835 0,838	0,865 0,837 0,840	0,865 0,837 0,840	0,850 0,824 0,827
Arbre de décision (AD)	0,801 0,796 0,797	0,797 0,793 0,794	0,808 0,801 0,802	0,795 0,790 0,791	0,802 0,796 0,797
Descente de gradient stochastique (DGS)	0,824 0,807 0,809	0,812 0,802 0,803	0,775 0,775 0,775	0,813 0,800 0,802	0,804 0,794 0,796
Vote CBN+MVS+RL	0,867 0,837 0,840	0,864 0,836 0,839	0,864 0,836 0,839	0,860 0,828 0,832	0,853 0,825 0,828

TABLE 3 : résultats obtenus sur le corpus de développement en précision (1re ligne), rappel (2e ligne) et f1-mesure (3e ligne)

3 Résultats sur les données de test

A partir des résultats obtenus dans la section précédente, nous avons sélectionné les 4 meilleures combinaisons de prétraitements et modèle :

- PT1 + Machine à vecteurs de support ;
- PT1 + Régression logistique ;
- PT3 + Régression logistique ;
- PT3 + Machine à vecteurs de support.

En plus de ces combinaisons, nous avons ajouté un *run* de vote majoritaire entre les 3 meilleurs classifieurs sur le prétraitement PT3 :

- Vote PT3 + {Classification bayésienne naïve, Machine à vecteurs de support, Régression logistique}

Les résultats qui nous ont été rendus par les organisateurs de la tâche sont présentés dans la Table 4. Après réception des résultats, nous avons réalisé que notre système de vote n'était pas correct car il prenait en compte des pondérations approximatives fournies par les classifieurs. C'est pourquoi nous ajoutons une dernière ligne dans les résultats, qui ne fait pas partie des résultats officiels mais qui correspond à un vote majoritaire corrigé dont les résultats ont été calculés après la fin de la campagne.

Modèle	Précision	Rappel	F1-mesure
PT3 + RL	0,806	1,000	0,893
PT1 + RL	0,806	1,000	0,893
PT1 + MVS	0,800	1,000	0,889
PT3 + MVS	0,799	1,000	0,888
Vote PT3 + CBN+MVS+RL	0,792	1,000	0,884
Vote PT3 + CBN+MVS+RL (correctif)	*0,806*	*1,000*	*0,893*

TABLE 4 : résultats sur le corpus de test *(la ligne en italique, correction du système de vote, ne fait pas partie des résultats officiels)*

Face aux très bons scores en rappel, il est intéressant de connaître le nombre de tweets retournés par catégorie. Afin de comparer plus en détails les résultats entre classifieurs ainsi qu'avec le référentiel, la Table 5 fournit le nombre de tweets pour chaque classe.

Corpus	#Transport	#Non-transport
PT1 + MVS	4 756	3 060
PT1 + RL	4 684	3 132
PT3 + RL	4 663	3 153
PT3 + MVS	4 714	3 102
Vote PT3 + CBN+MVS+RL	4 918	2 898
Vote PT3 + CBN+MVS+RL (correctif)	4 723	3 093
Test	***3 941***	***3 875***

TABLE 5 : nombre de tweets trouvés par classe pour chacun des classifieurs

Tous nos classifieurs, sans exception, ont privilégié la classe transport. Ceci est sans doute dû à un léger sur-apprentissage de cette classe.

Les résultats du classifieur utilisant la régression logistique sont plus hauts que ceux du classifieur utilisant la machine à vecteurs de support. La différence entre les prétraitements (c.-à-d. simple tokenisation et suppression des URLs *vs* l'ajout des suppressions de la casse, de la ponctuation et des mots vide) est ténue et montre *a priori* que les classifieurs ne nécessitent pas d'effectuer de prétraitements de base pour viser une quelconque amélioration de leurs performances.

Il faut par ailleurs noter que le vote majoritaire n'améliore pas les résultats, bien qu'il confirme ceux du classifieur le plus performant.

4 Conclusion

Dans cet article nous présentons les travaux réalisés par Syllabs sur la première tâche de la campagne DEFT 2018. Ceux-ci ont été réalisés en très peu de temps, notre inscription, la première à une campagne DEFT, s'étant faite tardivement. Pour autant, nous avons obtenus des résultats état de l'art qui sont proches des meilleurs classifieurs ayant participés à cette campagne.

Nous sommes restés sur des approches classiques de classification supervisées. Sans être exceptionnels, les scores en f1-mesure sont hauts, même si cela est essentiellement dû aux scores en rappel à 1. En étudiant un peu plus en détails les résultats ainsi que les différents paramètres des classifieurs, nous pourrions sans doute augmenter légèrement la précision. De plus, c'est sans doute l'apport de données extérieures comme des lexiques ciblés sur le transport, ou la correction des tweets qui auront tendance à améliorer les performances.

Toutefois, l'un des faits les plus marquants de nos travaux est le faible impact des prétraitements sur ces résultats. Au final, l'utilisation presque originale des tweets donne des résultats convaincants, sans trop d'effort. Nous pensons par la suite étudier plus en détails ces résultats, à commencer par l'analyse de l'impact des prétraitements.

Références

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L., TORRES-MORENO J.-M. (2018) DEFT2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en Île de France. In: Actes de DEFT. Rennes, France.

LOPER E. AND BIRD S. (2002) "NLTK: The Natural Language Toolkit," in Proc. ACL Workshop Effective Tools Methodologies Teaching Natural Language Process. Comput. Linguistics, 2002.

LIRMM@DEFT-2018 – Modèle de classification de la vectorisation des documents

Waleed Mohamed Azmy[1] Bilel Moulahi[1, 2] Sandra Bringay[1] Jérôme Azé[1], Maximilien Servajean[1]
(1) LIRMM, Université de Montpellier, CNRS, Montpellier, France
(2) IUT de Béziers, Université de Montpellier, France
prenom.nom@lirmm.fr

RÉSUMÉ

Dans ce papier, nous décrivons notre participation au défi d'analyse de texte DEFT 2018. Nous avons participé à deux tâches : (i) classification transport/non-transport et (ii) analyse de polarité globale des tweets : positifs, negatifs, neutres et mixtes. Nous avons exploité un réseau de neurone basé sur un perceptron multicouche mais utilisant une seule couche cachée.

ABSTRACT

LIRMM DEFT-2018 – Document Vectorization Classification model

In this paper, we describe our participation to the DEFT 2018 French Text Mining Challenge. The goal of the challenge is the sentiment analysis of French tweets. We participated to two tasks : (i) Transport/non-transport classification and (ii) Polarity analysis of positive, negative, neutral and mixed sentiments. We explored a neural network based on MultiLayer Perceptron using only one hidden layer.

MOTS-CLÉS : Analyse de polarité, réseaux de neurone, word embedding, doc2vec.

KEYWORDS: Polarity analysis, neural networks, word embedding, doc2vec.

1 Modèle de l'équipe ADVANSE du LIRMM pour l'édition 2018 de DEFT

De nombreux modules de traitement du langage naturel (NLP) commencent par l'extraction de certaines caractéristiques importantes du texte. Ces caractéristiques peuvent être, par exemple, le nombre ou la fréquence de mots spécifiques, des motifs prédéfinis, l'étiquetage grammatical, etc. Ces caractéristiques sont souvent définies manuellement et doivent être choisies avec soin, voire même nécessiter l'intervention d'un spécialiste des données étudiées. Bien que des résultats intéressants puissent être obtenus avec de telles approches, l'un des inconvénients récurrents est souvent la faible capacité à généraliser.

Depuis quelques années, plusieurs approches proposent d'utiliser des méthodes de vectorisation de mots et de documents. Ces stratégies qui convertissent des mots, des phrases ou même des documents entiers en vecteurs prennent en considération l'ensemble du texte et pas seulement certaines de ces parties. Il existe de nombreuses façons de transformer un texte en un espace à haute dimension comme la fréquence des termes et la fréquence inverse des documents (TF-IDF), l'analyse sémantique latente

(LSA), l'allocation de Dirichlet latente (LDA), etc(Maas *et al.*, 2011).

Cette nouvelle approche a été révolutionnée par Mikolov et al(Mikolov *et al.*, 2013a,b) qui a proposé le Continuous Bag Of Words (CBOW) et les modèles de sauts de grammaires connus sous le nom de Word2Vec. Il s'agit d'un modèle probabiliste qui utilise une architecture de réseau de neurones à deux couches pour calculer la probabilité conditionnelle d'un mot compte tenu de son contexte. Sur la base de ces travaux, Le et al. proposent un modèle vectoriel de paragraphe.

L'algorithme, également connu sous le nom de Doc2Vec, apprend les représentations de longueur fixes à partir de textes de longueur variable, tels que des phrases, des paragraphes et des documents(Le & Mikolov, 2014). Les vecteurs de mots et les vecteurs de documents sont formés par les modèles de langage de gradient stochastique et de rétro-propagation des réseaux neuronaux.

Tout d'abord, nous avons formé un modèle Doc2Vec sur l'ensemble du corpus d'entraînement fourni par les organisateurs du défi. Chaque tweet est traité comme un document séparé. Après avoir construit le modèle de vectorisation du document, chaque tweet peut être représenté avec un vecteur de N caractéristiques. Les principaux paramètres de construction d'un tel modèle sont donnés dans le tableau 1. Nous faisons varier le nombre de dimensions de 100 à 400 et essayons d'optimiser le modèle en utilisant la validation croisée et la mesure de précision. Pour les deux tâches, le nombre de dimensions pouvant représenter un tweet était égal à 250.

Paramètre	**Valeur**
Learning Rate	0.001
Nombre de dimensions utilisées	de 100 à 400
Context Window Size	10
Training epochs	20
Loss	Negative Sampling
Minimum word count	2

TABLE 1 – Document Vectorization Main Parameters

Ces vecteurs sont utilisés comme ensemble d'apprentissage pour entraîner un second réseau neuronal basé sur un perceptron multicouches. Pour la première tâche, deux classes ont été utilisées, alors que pour la seconde tâche, nous avons utilisé quatre classes. Nous avons également utilisé la descente de gradient stochastique et le réseau neuronal de rétro-propagation avec une couche cachée de 150 neurones.

2 Résultats

Les résultats des deux tâches, ainsi que quelques statistiques sur les autres soumissions sont présentés dans le tableau 2. La micro moyenne F1-Mesure est utilisée pour évaluer les expérimentations. Il y a 39 soumissions pour la première tâche et 41 pour la deuxième tâche. Les résultats montrent que le modèle n'arrive pas à prédire l'information globale et ne prête pas attention aux sentiments. Le résultat de la première tâche semble être meilleur, mais en général, les modèles devraient être plus profonds pour que nous puissions obtenir de meilleures performances.

	Notre modèle	Moyenne	Déviation Standard	Min	Max
Task-1	0.827	0.89	0.032	0.719	1
Task-2	0.38	0.727	0.162	0.38	1

TABLE 2 – Micro-mean F1 measure for the proposed model and statistics from other models

3 Conclusion et discussion

Notre modèle essaie simplement de faire une classification sans dictionnaires ou caractéristiques spécifiquement créées pour la tâche d'intérêt. Notre travail peut être vu comme une première étape en essayant de donner une réponse à la question ouverte : "Devrions-nous nous préoccuper de la linguistique ?" Nous pensons clairement que la réponse fournie par ce travail préliminaire est "oui".

L'utilisation des modèles CBOW et Skip-grams pour vectoriser le texte pourrait être bénéfique, mais l'inclusion de certaines caractéristiques du dictionnaire ou de signaux d'attention peut aider. Une autre façon est de construire le deuxième modèle en apprenant beaucoup plus en profondeur et non en utilisant seulement un réseau neuronal plat. Certains modèles tels que Convolution Neural Networks (CNN) ou Recurrent Neural Networks (RNN) permettent de pousser le modèle à aller plus loin et à prendre en considération les sentiments.

Remerciements

Nous remercions la région Occitanie et l'Agglomération Béziers Méditerranée qui finance la thèse de Waleed Mohamed Azmy, ainsi que la Fondation FondaMental qui finance le contrat d'ingénieur de recherche de Bilel Moulahi.

Références

LE Q. V. & MIKOLOV T. (2014). Distributed representations of sentences and documents. *CoRR*, **abs/1405.4053**.

MAAS A. L., DALY R. E., PHAM P. T., HUANG D., NG A. Y. & POTTS C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, volume 1.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, **abs/1301.3781**.

MIKOLOV T., YIH S. W.-T. & ZWEIG G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT-2013)* : Association for Computational Linguistics.

Adapted Sentiment Similarity Seed Words For French Tweets' Polarity Classification

Amal Htait[1, 2]
(1) Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France.
(2) Aix Marseille Univ, Avignon Université, CNRS, EHESS, OpenEdition Center, Marseilles, France.

ABSTRACT

We present, in this paper, our contribution in DEFT 2018 task 2 : "Global polarity", determining the overall polarity (Positive, Negative, Neutral or MixPosNeg) of tweets regarding public transport, in French language. Our system is based on a list of sentiment seed-words adapted for French public transport tweets. These seed-words are extracted from DEFT's training annotated dataset, and the sentiment relations between seed-words and other terms are captured by cosine measure of their word embeddings representations, using a French language word embeddings model of 683k words. Our semi-supervised system achieved an F1-measure equals to 0.64.

RÉSUMÉ

Mots-graines de Similarité de Sentiment Adaptés pour la Classification de Polarité des Tweets en Langue Française.

Cet article présente notre contribution en DEFT 2018 tâche 2 : "Polarité globale", déterminant la polarité globale (Positif, Négatif, Neutre ou MixPosNeg) des tweets concernant les transports publics, en langue Française. Notre système est basé sur une liste de mots-graines de sentiment adaptés aux tweets de transport public français. Ces mots-graines sont extraits de corpus annoté de DEFT, et les relations entre les mots-graine et les autres termes sont capturées par la similarité en mesure de cosinus entre les vecteurs représentants les mots, en utilisant un modèle word2vec en langue Française de 683k mots. Notre système semi-supervisé a atteint un F1-measure égale à 0,64.

KEYWORDS : Seed-words, Twitter, Similarity Measures, Word Embeddings, Word2vec.

MOTS-CLÉS: Mots-graines, Twitter, Mesure de la Similarité, Plongement de mot, Word2vec.

1 Introduction

Sentiment Analysis aims to obtain feelings expressed as positive, negative, neutral, or even expressed with different strength or intensity levels. One of the well known extracting sentiment approaches is the lexicon-based approach. A sentiment lexicon is a list of words and phrases, such as $excellent$ and $awful$, each is being assigned with a sentiment polarity. Using sentiment lexicon can provide rich sentiment information what make it the foundation of many sentiment analysis systems (Liu, 2012).

In our previous work (Htait *et al.*, 2017a), (Htait *et al.*, 2017b), we used tweet's adapted seed-words in English and Arabic languages as sentiment lexicon, and then we extracted a score for our test sentences based on the cosine similarity measure between their word embeddings vectors and the sentiment seed-words word embeddings vectors. Based on that score, the sentences were classified

positive, negative or neutral. To participate in DEFT 2018 challenge (Paroubek *et al.*, 2018), we had to adapt our method to DEFT dataset as language, adapted seed-words and number of classes since DEFT classify tweets as Positive, Negative, Neutral and $MixedPosNeg$.
The detailed description of the system and the results of our participation in DEFT 2018 are presented in the sections below.

2 Related Work

The seed-words were the base of many sentiment analysis experiments, some used the concept of seed-words with supervised or semi-supervised methods. For example, Ju et al. (Ju *et al.*, 2012) worked on a semi-supervised method for sentiment classification that aims to train a classifier with a small number of labeled data (called seed data). Some other experiments used the concept with unsupervised methods which reduces the need of annotated training data. For example Turney (Turney, 2002; Turney & Littman, 2003), which used statistical measures, such as point wise mutual information (PMI), to calculate the similarities between words and its list of 14 sentiment seed-words ($good$, $nice$, $excellent$, $positive$, $fortunate$, $correct$, $superior$, bad, $nasty$, $poor$, $negative$, $unfortunate$, $wrong$, $inferior$).

In our previous work (Htait *et al.*, 2017a), new lists of adapted seed-words were extracted in English language. The most frequent words in Sentiment140 (Go *et al.*, 2009) were retrieved and then manually filtered to eliminate the irrelevant words. The tests in (Htait *et al.*, 2017a) showed the efficiency of the new seed-words over Turney's 14 seed-words in sentiment polarity and sentiment intensity detection. Also, they showed that using cosine similarity measure of word embeddings representations (word2vec) yields better results than using statistical measures like PMI to calculate the similarities between words.

For the new system in French language, a similar method of our previous work is followed. A list of sentiment seed-words adapted to tweets regarding french public transport is created, and the fourth type of sentiment classification (MixPosNeg) is added to our system.

3 Adapted Seed-words

The seed-words are words with strong semantic orientation, chosen for their lack of sensitivity to the context. They are used as paradigms of positive and negative semantic orientation. Adapted seed-words are seed-words with the characteristic of being used in a certain context or subject. In our previous work (Htait *et al.*, 2017a),(Htait *et al.*, 2017b), the extracted seed-words were adapted to micro-blogs in general. For example, the word $cool$ is an adjective that refers to a moderately low temperature and has no strong sentiment orientation, but it is often used in micro-blogs as an expression of admiration or approval. Therefore, in micro-blogs, the word $cool$ is considered a positive seed-word and the tweet including it is mostly considered a positive tweet.
For DEFT 2018 challenge, and since the tweets are about the french public transport, the extracted seed-words are adapted to this subject. For example the word $retard$ (as $late$ in French) is considered a negative seed-word since the tweets are about transport, and a late train or bus usually provoke negative feelings.
The procedure of extracting seed-words is done in three steps :

1. Two lists of positive and negative tweets are created based on the DEFT 2018 training corpora of public transport tweets in French language.
2. The most frequent words in the two lists of tweets are extracted, after eliminating stop-words. As a result, two lists (positive and negative) of most frequent words are created.
3. A manual filter is applied on the two lists of most frequent words to eliminate the irrelevant words from the lists.

The list of French seed-words adapted to public transport tweets is as shown in Table 1, with 63 positive seed-words and 63 negative seed-words.

Positive	**Negative**
mdr, bien, merci, bon, mdrr, bonne, mdrrr, rigole, ptdr, juste, aime, beau, cool, super, coucou,respect, heureusement, rire, adore, bravo, mdrrrr, jolie, belle, blague, ok, gagner, ptdrrr, ptdrrrrr, ptdrr, génial, gnial, ouais, meilleur, bisous, courage, vive, offre, joie, haha, sourire, ptdrrrr, tranquille, gentil, parfait, mddr, bonheur, magnifique, jéme, jme, prfre, chou, mignon, gratuit, amour, bons, content, remercie ahah, ouf, direct, trql, heureux, mdrrrrr	retard, merde, grve, grave, ratp, putain, problème, trafic, problme, pute, puent, problèmes, odeur, coup, panne, mal, flemme, fdp, marre, problmes, gueule, bordel, accident, rater, con, chier, couilles, retards, perdu, pue, rat, casse, pu, grves, graves, bah, taper, bizarre, louper, loup, franais, travaux, galre, galère, fou, chiant, gnant, incident, galérer, peine, ptn, chelou, perdre, foutre, morte, tard, horrible, mauvais, loin, manque, connard honte, tape,

TABLE 1 – The Lists of Positive and Negative Adapted Seed-words.

For further tests, we extended the lists of adapted seed-words using NormAFE [1], a tool that creates dictionaries for micro-blogs normalization, in a form of pairs of misspelled word with its standard-form word, in the languages : Arabic, French and English. Using NormAFE, we extract the misspellings of our seed-words and we add them to the original list. For example some of the misspellings of the word $magnifique$ (as $magnificent$ in French) are : $magnfique$, $magnif$, $magnifi$, $magnifiiiique$, $magnifiiique$, $magnifiique$, $magnifik$, $magnifike$, $magnifiq$, etc. The result of this procedure is a 1358 positive seed-word and 1330 negative seed-words.

4 System of Sentiment classification

The System is based on sentiment similarity cosine measure with Word Embeddings representations (word2vec). For this purpose, the word embeddings model [2] of 48M French tweets and 683k words by (Htait *et al.*, 2018) is used.

To predict the tweets polarity, first, each tweet is cleaned by removing links, user names, stop words, numeric tokens and characters. Also most emoticons and emojis, like : " :-)" or ☹, are replaced by $positive\_emoji$ and $negative\_emoji$, since these expressions replace most emoticons and emojis in the word embeddings model. After that, the tweet is segmented into tokens or words. The similarity between each word with positive seed-words and negative seed-words is calculated using gensim tool [3] with the previously mentioned word2vec model. Having the sentiment score of each word in a

1. https ://github.com/amalhtait/NormAFE
2. https ://github.com/amalhtait/NormAFE/blob/master/Models/Note
3. https ://pypi.python.org/pypi/gensim

tweet, we aggregate by sum to combine these values. The final score specify the tweet's polarity as Positive, Negative or Neutral.

The DEFT 2018 task 2 requires determining the overall polarity of a tweet as Positive, Negative, Neutral or MixPosNeg. Therefore, we need to detect the tweets of mixed polarity as MixPosNeg, which is not covered by the current system. By observing the training dataset, we notice that a large number of mixed polarity tweets are a combination of a text with a certain polarity and an emoji with an opposite polarity, like the following example where the person is mostly complaining about negative events and then he adds a smiley at the end of his tweet which shows sarcasm and the expression of mixed sentiment : " ... des vieux types mn clc dans le métro et un pigeon s'est lacher sur ma veste ... ☺".

Based on the previous observation, we decide to consider the tweets of a certain polarity with an emoji of opposite polarity as a MixedPosNeg tweets.

5 Results

For DEFT 2018 task 2 challenge, four runs are sent to predict the polarity of 7816 public transportation tweets in French language :

- **Run_1** has the results of the system by using the extended seed-words of Table 1 (1358 positive and 1330 negative), and without adding the fourth class of MixPosNeg prediction. Therefore, the results contain only three classes Positive, Negative and Neutral.
- **Run_2** has the results of the system by using the extended seed-words of Table 1 (1358 positive and 1330 negative), with the fourth class of MixPosNeg prediction added to the results.
- **Run_3** has the results of the system by using the 126 seed-words of Table 1, and without adding the fourth class of MixPosNeg prediction. Therefore, the results contain only three classes Positive, Negative and Neutral.
- **Run_4** has the results of the system by using the 126 seed-words of Table 1, with the fourth class of MixPosNeg prediction added to the results.

The Table 2 shows the official results of DEFT 2018 task 2 challenge, where the Run_3 achieved an F1-measure equals to 0.64, as the best result between our four runs. The results show that using an extended version of the seed-words decreased the F1-measure from 0.64 in Run_3 to 0.62 in Run_1. Also, our method to predict the fourth class $MixedPosNeg$ decreased the F1-measure to 0.63 in Run_4 and to 0.61 in Run_2.

	F1-measure
DEFT Best Result	0.82288
Run_1	0.62539
Run_2	0.61622
Run_3	**0.64524**
Run_4	0.63939

TABLE 2 – Runs Result at DEFT 2018 Task 2 challenge - Global polarity of tweets regarding public transport, in French language.

6 Conclusion

This paper presents our contribution in DEFT 2018 task 2 : "Global polarity". The system used is based on a list of sentiment seed-words adapted for French public transport tweets. These seed-words are extracted from DEFT's training annotated dataset, and the sentiment relations between seed-words and other terms are captured by cosine measure of their word embeddings representations, using a word embeddings model of 683k French words. We participated at the DEFT challenge with four runs, and our best run achieved an F1-measure equals to 0.64 as shown in Table 2.

Even though our results were not the best, but they are promising results since our best results are achieved predicting only three classes : Positive, Negative and Neutral, in a challenge where the prediction of four classes is required (Positive, Negative, Neutral and MixedPosNeg). Unfortunately, our method to predict the fourth class $MixedPosNeg$ decreased the F1-measure to 0.63 in Run_4 and to 0.61 in Run_2. Therefore, and as a future work, we are seeking on a new method to predict the fourth class $MixedPosNeg$, by predicting the polarities of tweet segments and detecting opposite polarities in the same tweet.

Acknowledgments

This work has been supported by the French State, managed by the National Research Agency under the «Investissements d'avenir»program under the EquipEx DILOH projects (ANR-11-EQPX-0013)

References

GO A., BHAYANI R. & HUANG L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, **1**(12).

HTAIT A., FOURNIER S. & BELLOT P. (2017a). Identification automatique de mots-germes pour l'analyse de sentiments et son intensité. In *CORIA - RJCRI*, Marseille, France.

HTAIT A., FOURNIER S. & BELLOT P. (2017b). Lsis at semeval-2017 task 4 : Using adapted sentiment similarity seed words for english and arabic tweet polarity classification. *In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, p. 718–722.

HTAIT A., FOURNIER S. & BELLOT P. (2018). Unsupervised creation of normalisation dictionaries for micro-blogs in arabic, french and english. In *International Conference on Computational Linguistics and Intelligent Text Processing*.

JU S., LI S., SU Y., ZHOU G., HONG Y. & LI X. (2012). Dual word and document seed selection for semi-supervised sentiment classification. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, p. 2295–2298 : ACM.

LIU B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, **5**(1), 1–167.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L. & TORRES-MORENO J.-M. (2018). Deft2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en île de france. In *Actes de DEFT*, Rennes, France.

TURNEY P. D. (2002). Thumbs up or thumbs down ? : semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, p. 417–424 : Association for Computational Linguistics.

TURNEY P. D. & LITTMAN M. L. (2003). Measuring praise and criticism : Inference of semantic orientation from association. *ACM*, **21**(4), 315–346.

ATALA

ARIA

UMR IRISA

Inria
INVENTEURS DU MONDE NUMÉRIQUE

orange

UNIVERSITE BRETAGNE LOIRE MATHSTIC

METROPOLE Rennes
vivre en intelligence

INSA RENNES

Université Bretagne Sud
ubs

fondation RENNES 1

sylLabs

Script&Go

edp sciences

Qwant

Synapse